How Far Can Instruction Steering Go? Coverage, Quality, and Calibration

Anonymous authors
Paper under double-blind review

Abstract

Activation steering promises modular control of language models: learn one direction per instruction, then add the directions that a request needs. This promise rests on two assumptions: directions that work alone also work together, and an answer that passes an instruction checker is still a good answer. We test both assumptions on seven models. We calibrate one edit per instruction and freeze it. An answer counts as a success only if it passes every checker and a blind quality judge, which agrees with a human audit. We separate *installation*, where the edit must carry an instruction that the prompt omits, from *reinforcement*, where the prompt already states it. Few instructions can be installed reliably. The quality judge shrinks this set further. On four smaller models, it removes all four installations that the automatic checks accept, and it roughly halves the usable reinforcements. Where installations do succeed alone, as on Gemma-3-27B, they also compose. On instruction pairs, they beat the unedited model by 35.0 points. Still, they trail simply stating the instructions by 17.2 points. Reinforcement gives mixed results and falls behind prompting as sets grow, yet much of this loss comes from calibration. Composing steering directions is therefore feasible, but its value depends on coverage, answer quality, and calibration, and prompting remains the baseline to beat.[1]

1 Introduction

A useful answer often has to satisfy several requirements at once. A model may need to answer the question while also following constraints on format, style, length, or wording. Instruction following benchmarks capture this setting through prompts with multiple verifiable constraints (Zhou et al., 2023; Jiang et al., 2024). Activation steering offers a modular way to provide such control: identify an internal direction for a behavior and intervene on that direction at inference time (Turner et al., 2024; Rimsky et al., 2024; Zou et al., 2025; Li et al., 2023; Jorgensen et al., 2023). Work on task and function vectors further suggests that some behaviors can be represented compactly and reused across contexts (Hendel et al., 2023; Todd et al., 2024). This motivates a simple goal: **learn one direction per instruction and combine the directions needed for a new request**.

Single-instruction success, however, does not guarantee useful control. A checker can confirm that an output has a title, uses lowercase text, or ends with a required phrase. It cannot tell us whether the answer is still relevant, coherent, or useful. Instruction following work therefore treats constraint satisfaction and response quality as distinct evaluation problems (Jiang et al., 2024). Activation interventions show the same tension: stronger steering can improve the target behavior while reducing helpfulness or generation quality (Li et al., 2023; Scalena et al., 2024). Constraint compliance alone can therefore overstate steering success.

Composition adds a second challenge. A direction that works alone may change with intervention strength, construction, layer choice, prompt context, or the presence of other edits (Jorgensen et al., 2023; Scalena et al., 2024; Adila et al., 2024). Prior work shows that internal vectors can sometimes compose (Todd et al., 2024; Ilharco et al., 2023; Scalena et al., 2024; Radevski et al., 2026; Kang et al., 2026). These results suggest a natural expectation: directions that work alone should add up

[1] Code: `https://anonymous.4open.science/r/ISC`

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.

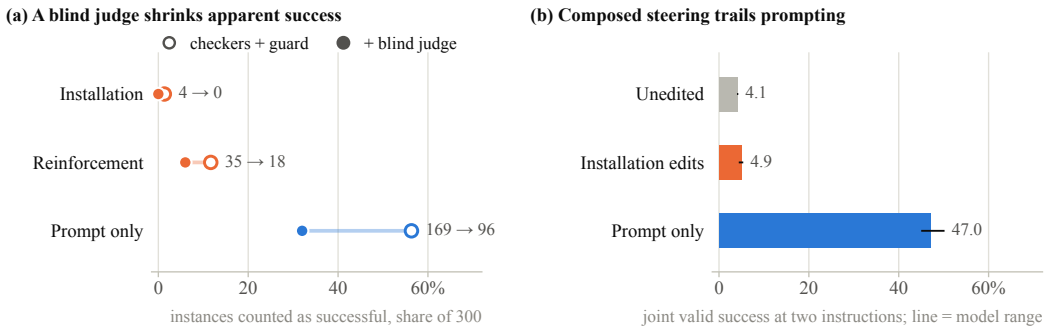


Figure 1: **Checker success overstates instruction steering, and composed steering trails prompting.** (a) Successful instances per condition on the four smaller models (Table 1). In installation, steering carries an instruction the prompt omits; in reinforcement, it supports an instruction the prompt states. Reinforcement counts only instances that the prompt alone does not already satisfy. (b) Joint valid success at two instructions, on sets drawn from all 25 instructions, averaged over three models (Table 3). An answer counts only if it satisfies every instruction and passes both the guard and the judge.

when combined. We find that this holds only in part (Figure 1). Once answer quality is required, few directions work alone, and those that do compose but still trail simply stating the instructions.

To test this expectation, we treat each individually calibrated direction as a *frozen module* and reuse it without recalibrating for the final instruction set. We evaluate both constraint satisfaction and instruction-blind answer quality, following prior concerns about steering quality and accumulated interventions (Stolfo et al., 2025; Subbiah et al., 2026; Wu et al., 2025). We distinguish *installation*, where steering carries an instruction absent from the prompt, from *reinforcement*, where steering supports an instruction already stated. With this setup, we can trace failures to limited coverage, quality loss, calibration, or the composition rule. We make three main contributions:

1. **Checker compliance overstates steering success.** We score test answers with the instruction checkers and the guard, and we judge guard-valid answers with an instruction-blind quality judge. The judge agrees with an audit by 3 human raters on 80.0% of audited answers ($\kappa = 0.60$). On four smaller models, the judge removes all 4 usable installations and cuts usable reinforcements from 35 to 18 (Sections 3 and 7).
2. **Directions that succeed alone do compose, but few succeed alone.** Most instructions have no usable installation, so composing the full pool stays close to the unedited model. Where members succeed alone, on Gemma-3-27B, their composition beats the unedited model by 35.0 points on pairs but trails prompting by 17.2. Much of the reinforcement loss comes from calibration. At four instructions, per-set dose tuning recovers 4.3 to 20.0 points on the three composition models, and a different composition rule avoids much of the frozen loss (Section 4).
3. **Prompt-side methods set the bar for steering.** Explicit prompting beats installation edits, and a generic retry after a failed check raises joint valid success by 3.9 to 5.6 points. Detailed checker feedback adds a resolved gain on only one of three models (Section 5).

2 Experimental Design and Measurements

Tasks and data. An *instruction instance* specifies a constraint and its arguments, such as a required closing phrase. We use the 25 constraint types and strict deterministic checkers from IFEval (Zhou et al., 2023), with three instances per type on the smaller models. For transfer, we use 20 IFBench constraint types with one instance each (Pyatkin et al., 2025). Base tasks are open-ended English writing requests from No Robots (Rajani et al., 2023), Dolly (Databricks, 2023), and OpenAssistant (Köpf et al., 2023). We retain tasks with no existing length, count, or format constraint and whose human reference answer contains at least 250 words. Disjoint prompt-identity splits contain 400 extraction, 300 development, 400 test, and 1,000 held-out comparison prompts.

Models. Single-instruction experiments use Qwen2.5-7B-Instruct (Qwen, 2025), Gemma-3-4B-IT (Gemma Team, 2025), Llama-3.1-8B-Instruct (Grattafiori et al., 2024), and Phi-3-mini-4k-Instruct (Abdin et al., 2024). We also evaluate Qwen2.5-32B-Instruct (Qwen, 2025), Gemma-3-27B-IT

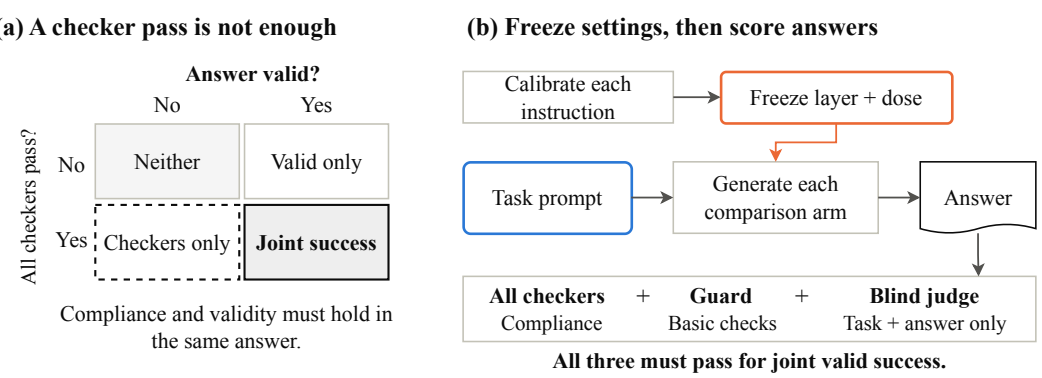


Figure 2: **Problem and evaluation setup.** *Left:* constraint compliance and answer quality are separate, and joint success requires both. The dashed cell shows checker success with an invalid answer. *Right:* we calibrate edits on single instructions, freeze their settings, and compare prompts that state or omit the instructions with and without steering. Joint success requires all checkers, the automatic guard, and the blind judge, which sees only the task and answer.

(Gemma Team, 2025), and Llama-3.1-70B-Instruct (Grattafiori et al., 2024). For these larger models, we use one instance per constraint, one extraction seed, and 100 test prompts (400 in the Gemma-3-27B installation study). Phi-3 is excluded from composition experiments. We use greedy decoding with each model's chat template. Token limits are calibrated by model and response-language instance. Exact repetition loops stop early and count as cut off (Appendix A).

Installation vs. reinforcement. In *installation*, the prompt contains only the base task, so the edit must carry the target instruction. In *reinforcement*, the prompt also states the instruction, so the edit supplements information already present in context. We calibrate these conditions separately (Figure 2). For composition, we compare four arms: the base task alone, installation edits without instruction text, the instruction list alone, and the instruction list with layer-separated reinforcement edits. We call the final arm the *hybrid*. We evaluate priority control separately as an extended reinforcement experiment (Appendix M).

Validity and joint success. Let $C_i(y)$ indicate whether answer y passes instruction i's strict checker. The automatic guard $G(x, y)$ requires at least 20 words, no more than half of the answer's 4-grams to repeat, and no mid-sentence truncation at the token limit. We call failure of either of the first two conditions *degeneration*. Our primary quality judge is Claude Sonnet 5. The judge sees only the base task x and the answer y. It decides whether the answer addresses the task, and it ignores constraints on format, style, case, length, language, and wording. Let $Q(x, y)$ indicate a yes verdict; no, unsure, and refusal count as failures. We define full validity as $V(x, y) = G(x, y)Q(x, y)$ and report both guard validity and full validity. The full rubric appears in Appendix K. A human audit agrees more with a combined quality criterion than with task relevance alone (Section 7). Neither the judge nor the audit establishes factual correctness. A second judge, GPT-5.6-terra, agrees with the primary judge on 84.7% to 89.0% of prompt-only and hybrid composition answers on the three smaller composition models (Appendix K). For a set S of K instructions, the outcomes are

$$p_i^{\text{valid}} = \mathbb{E}[V(x, y)C_i(y)], \qquad J_S = \mathbb{E}\left[V(x, y)\prod_{i \in S} C_i(y)\right]. \tag{1}$$

Here, J_S is *joint valid success*: an answer counts only if it is valid and satisfies every instruction in S. Guard-only versions replace V with G, and invalid answers remain in the denominator. Low joint success does not by itself imply interference: if independent instructions each succeed with probability 0.8, joint success falls to about 3.5% at $K = 15$.

Extraction and calibration. At each candidate layer, a steering direction is the normalized difference between two mean activations at the last input token: one with the instruction and one without it. We compute it on extraction prompts only. On development prompts, we first search seven layers and six coarse doses, and select one layer per constraint and prompt condition. We then search a finer dose grid and freeze one dose per instance. We select the layer with the most usable coarse doses. The frozen dose is the midpoint of the longest usable interval on the fine grid. Fallback rules maximize guard-qualified compliance within the quality bounds. Calibration uses only the auto-

Task: Who are the best starting MLB pitchers of all time?

Instructions: Finish your response with this exact phrase Is there anything else I can help with?. No other words should follow this phrase. Your entire response should be in English, and in all capital letters.

Answer (Qwen2.5-7B, prompt plus reinforcement edits): `THE BEST STARTING MLB PITCHERS OF ALL TIME INCLUDES GIBSON KEN, SPITZER RYAN, AND COCHRAN JOE. IS THERE ANYTHING ELSE I CAN HELP WITH?`

Scores: both checkers and the guard pass; the blind judge rejects; all three raters report wrong facts.

Figure 3: **Checker success does not mean a useful answer.** A real answer from the human audit (Appendix K.2). It satisfies both instructions and the guard, but the pitcher names are garbled. The blind judge and the human raters both reject it.

matic guard: the quality judge is applied after freezing and never triggers reselection. A setting is *guard-qualified* if three conditions hold: $\mathbb{E}[G(x,y)C_i(y)] \geq 0.80$, guard validity drops by at most five percentage points from the unedited arm, and degeneration rises by at most five points.

Frozen evaluation and classification. We evaluate frozen settings on 200 test prompts for the smaller models. We first test whether unedited compliance reaches 0.80, then whether the edit is guard-qualified. Installation additionally requires unedited compliance below 0.80, a gain of at least 20 percentage points, and a paired interval excluding zero. Thus, a usable installation must satisfy both guard qualification and this installation effect criterion. We separately classify gains confined to unacceptable outputs and cases with no adequate improvement. Appendix A gives the full rules.

How frozen settings compose. A relative edit e adds $\|h_{\text{last}}\|e$ to the residual stream at every prompt and generated token position. The reference norm is recomputed for each sequence at every forward call. For shared layer composition, we use the most common frozen member layer and the mean frozen dose s, and read all directions at that layer. Let D contain the unit direction vectors, with Gram matrix $G_D = D^\top D$ and coefficients w. Our normalized composition rules use $e = s\left(Dw \div \sqrt{w^\top G_D w}\right)$. Fixed-norm addition sets $w = \mathbf{1}$, while the calibrated sum preserves each member's frozen dose and uses $e = \sum_i s_i d_i$ without final normalization. Layer-separated composition instead keeps each member at its own frozen layer and dose. When members share a layer, their contributions are summed. The *hybrid* applies this rule to reinforcement settings. We also test inverse Gram correction, orthogonalized composition, and the activation-dependent rule of Stolfo et al. (2025). Appendix J gives their definitions, tuning choices, and additional controls.

Sets, tuning, and uncertainty. We admit a set only if three designated development prompts produce witness answers that pass all member checkers and the quality judge. We evaluate the full instruction pool, guard-qualified reinforcement members, and a separate set of judge-qualified installation members for Gemma-3-27B. The screen retains all 30 full-pool pairs but becomes more selective for larger sets (Appendix D). As a reference, we tune a per-set dose multiplier on development prompts, plus the ridge parameter where it applies. This tuning changes no member layer and uses no test outcome. Deployment experiments use all 200 test positions per set on the smaller models. The central held-out comparison uses a fixed subset of 200 of the 1,000 positions, seeded by set. Reference and revision experiments use 100 positions per set. Guard-invalid outputs remain failures. We compute intervals from 10,000 paired bootstrap samples over sets and prompts, weighting sets equally and sharing resamples across comparison arms (Appendix A). Additional repeat and extraction-seed controls appear in Appendix F.

3 Answer quality changes single-instruction feasibility

Composition depends on the quality of the individual directions it combines. We therefore test which single-instruction settings remain usable when answer quality must accompany checker compliance. This is stricter than asking whether steering changes the checker outcome at all. Figure 3 shows why: a steered answer can satisfy every checker and the guard yet fail the task. In the human audit, raters reject 46 of the 124 answers that pass every checker: 32 of 66 steered answers, but only 14 of 58 prompt-only answers (Appendix N).

The validity check changes which settings qualify. On the same frozen settings, adding the judge removes more than half of the usable settings (Table 1). Across the four smaller models, 4 of 300 instances have a usable installation under the guard alone, all on Llama-3.1-8B. With the judge, none do, and usable reinforcements fall from 35 to 18. The gap is not limited to steering: in the unedited prompted model, already-satisfied instances drop from 169 to 96. The guard's minimum length

Table 1: **Frozen single-instruction classifications under both validity rules.** Usable counts apply the success target and quality bounds. Satisfied counts refer to the unedited prompted model. Each row contains n instances per prompt condition. Adding the judge removes more than half of the usable settings. On the four smaller models, usable installations fall from 4 to 0 and usable reinforcements from 35 to 18. Appendix E reports threshold sensitivity.

		Usable installation		Usable reinforcement		Already satisfied	
Model	n	Auto. Guard	Guard & Judge	Auto. Guard	Guard & Judge	Auto. Guard	Guard & Judge
Qwen2.5-7B	75	0	0	5	5	51	26
Gemma-3-4B	75	0	0	9	5	49	37
Llama-3.1-8B	75	4	0	11	5	40	29
Phi-3-mini	75	0	0	10	3	29	4
Qwen2.5-32B	25	2	0	2	0	20	18
Gemma-3-27B	25	6	5	3	1	21	20
Llama-3.1-70B	25	3	0	2	1	20	18

Table 2: **Scoring all predefined test positions.** Number of instances reaching 80% compliance under each rule, out of 300 per condition across four smaller models. Each instance uses the same 200 positions across both arms and all rules; guard-invalid outputs fail. Counts include already-satisfied instances and omit usability bounds. Adding the judge removes the installation gain (43 vs. 38 becomes 34 vs. 35) and shrinks the reinforcement gain (204 vs. 169 becomes 110 vs. 96).

	Installation		Reinforcement	
Scoring rule	Unedited Instances	Edited Instances	Unedited Instances	Edited Instances
Loose checker	38	47	197	235
Strict checker	38	44	188	219
Strict + guard	38	43	169	204
Strict + guard + judge	35	34	96	110

conflicts with the constrained-response instruction, so we repeat the analysis without that type. The trend remains, with installations falling from 4 to 0 and reinforcements from 31 to 18 (Appendix E). Checker compliance alone can therefore overstate feasibility on our filtered writing tasks.

Full-position rescoring separates the three scoring effects. On the same outputs, strict checking, the guard, and the judge can each remove target-reaching cases. We rescore every test position across all 600 small-model frozen-test cells. Table 2 counts instances that reach the compliance target, without the additional usability bounds. For edited reinforcement, strict checking reduces the count from 235 to 219, the guard lowers it to 204, and the judge to 110. The unedited prompted arm falls in the same way, from 188 under strict checking to 169 with the guard and 96 with the judge.

Behavioral gains often remain below the usable target. Steering can produce a measurable behavioral change without reliable control. Under the guard, the tested search finds no adequate improvement for 239 of the 300 installation instances. Still, 22 pass the weaker installation effect criterion, with a median gain in valid compliance of 59.8 percentage points. Yet 18 of these 22 still lack a usable setting. Coverage stays limited at lower targets: on the four smaller models, at a judged target of 0.70, two installations become usable, both on Llama-3.1-8B, and at 0.50, four are usable across Gemma-3-4B and Llama-3.1-8B (Appendix E). Coverage is similarly limited on the 20 IFBench constraint types: calibration finds 0 usable installations across the 3 smaller composition models under the guard and 0 with the judge. On IFBench's own test prompts, adding the frozen reinforcement edits changes strict accuracy by -0.7 to 2.8 points, and every interval contains zero (Appendix O). Appendix C reports the outcome counts and selected dose curves.

Larger models improve coverage, with one exception. Each larger model yields more guard-qualified installations than its smaller family member on the matched instance. Adding the judge removes the usable installations on Qwen2.5-32B and Llama-3.1-70B, but retains 5 of 6 on Gemma-3-27B. On Qwen2.5-32B, the two installations fail the validity-loss bound despite high judged compliance. The small-model pattern therefore does not hold across all models: both the compliance target and the cost to answer validity shape coverage.

Calibration depends on the prompt condition. Reusing an installation setting when the prompt states the instruction helps little and can hurt. We test this on the 22 instances whose installation setting passes the effect criterion. Against prompting alone, reuse changes guard-valid compliance by a median of -2.3 points, and the largest loss is 57.5 points. Separately calibrated reinforcement

Table 3: **Two-instruction composition on the full pool.** Joint valid success (%) with 95% bootstrap intervals over 30 sets per model. Installation receives no instruction text; prompt only states the instructions, and the hybrid adds layer-separated reinforcement edits. Installation edits stay within 1.4 points of the unedited model, while prompt only adds 41 to 46 points.

Arm	Qwen2.5-7B	Gemma-3-4B	Llama-3.1-8B
Unedited	4.0 [0.0, 10.6]	4.3 [0.0, 10.9]	4.1 [0.0, 10.8]
Fixed-norm addition	4.0 [0.0, 10.6]	3.9 [0.0, 10.2]	4.4 [0.3, 11.2]
Layer-separated	5.4 [0.6, 12.1]	4.4 [0.5, 10.3]	4.9 [0.7, 11.7]
Prompt only	45.9 [35.8, 56.2]	50.2 [37.2, 63.1]	45.0 [33.0, 56.7]
Prompt plus the reinforcement edits (hybrid)	44.9 [34.6, 55.4]	52.8 [41.5, 64.0]	49.7 [38.8, 60.2]

Table 4: **Composing successful installations on Gemma-3-27B.** Joint valid success over all test prompts, with 95% bootstrap intervals across witness-admitted sets. Members use judge-qualified settings; the hybrid adds reinforcement edits to the prompt. At two instructions, every installation rule beats the unedited model, but prompting stays higher at both set sizes.

Arm	Two instructions included	Three instructions included
Unedited	42.9% [21.0, 64.2]	28.9% [7.2, 50.9]
Fixed-norm installation	56.3% [31.3, 77.7]	45.8% [22.4, 68.2]
Orthogonalized installation	54.8% [31.2, 75.8]	31.7% [11.6, 52.9]
Layer-separated installation	77.9% [71.8, 83.7]	46.1% [33.9, 58.1]
Prompt only	95.1% [93.2, 96.8]	95.1% [93.2, 96.8]
Prompt plus the reinforcement edits (hybrid)	92.4% [88.3, 95.4]	88.0% [81.6, 92.7]

settings add a median of 5.0 points on the same instances. This supports calibrating installation and reinforcement separately, but does not show that test-time retuning would solve composition.

4 Composition depends on coverage and calibration

We test composition in two settings. The full instruction pool measures practical coverage, while subsets of individually successful members test whether those directions remain reliable when combined. Reinforcement edits then test whether frozen settings still help when the prompt already states the instructions. Together, these analyses separate failures caused by limited single-instruction coverage from failures that emerge during composition.

Full-pool installation remains far below prompting. Most installation rules stay close to the unedited model, whereas prompting reaches 45.9%, 50.2%, and 45.0% on the three models (Table 3). No compatible small-model set contains only guard-usable installations. Qwen2.5-7B and Gemma-3-4B have none, and Llama-3.1-8B has two qualifying types that cannot be combined. The table thus uses arbitrary compatible sets, including members that fail individually. Poor calibration of the input-dependent coefficient's gain explains some output damage, but tuning reduces that damage without producing reliable installation. On Gemma-3-4B, the input-dependent coefficient at gain one yields 2.3% validity and 54.0% degeneration. Per-set development tuning raises validity to 23.1% and lowers degeneration to 13.4%, yet joint success is only 1.0% [0.0, 2.9]. Appendix D reports all rules and validity costs. Edits also cannot replace the prompt at scale: when the prompt omits four target instructions, injecting them, alone or among up to fifteen instructions, gives zero joint valid success on all three models, while, among prompts without edits, stating only the four targets gives the highest success. In this one fifteen-instruction set, adding the four target edits to that prompt raises target success further on Qwen and Llama (Appendix M).

Successful installations compose, but still trail prompting. On Gemma-3-27B, installations that each work alone keep their effect when combined (Table 4). This model provides 10 compatible pairs and 10 triples whose members each qualify under the judge. We use each member's calibrated instance, and every set passes the witness screen. Across all 400 test prompts, every rule beats the unedited model at two instructions, and all intervals exclude zero. The gains are 35.0 points [13.7, 57.8] for layer-separated composition, 13.4 [5.2, 25.0] for fixed-norm addition, and 12.0 [1.1, 28.8] for orthogonalized composition. The full pool could not reveal this effect because most of its members fail alone. Prompting remains stronger, exceeding layer-separated composition by 17.2 points [11.2, 23.5] at two instructions and 49.0 [36.5, 61.7] at three. At three instructions, the layer-separated gain over no edit is unresolved (17.2 [-8.5, 43.0]). The fixed-norm gain remains resolved (16.9 [4.6, 32.9]).

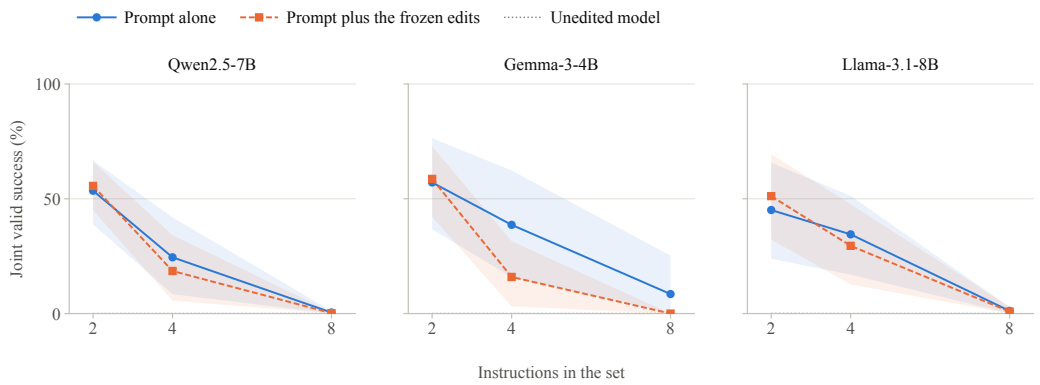


Figure 4: **Full-pool reinforcement weakens as sets grow.** Joint valid success on held-out prompts across 10 admitted sets per count and model. Blue circles show prompting. Orange squares add layer-separated reinforcement edits. The dotted line is the unedited model. Bands are per-arm 95% bootstrap intervals. At two instructions, hybrid-minus-prompt intervals include zero on all models; at four, the hybrid is lower on all three and excludes zero on Qwen and Gemma. Larger counts include only witness-admitted sets.

The quality criterion changes apparent reinforcement gains. The judge shrinks the hybrid's apparent lead over prompting. Under the guard alone, the hybrid exceeds prompting on all three smaller composition models: 61.3% versus 59.0%, 66.3% versus 56.7%, and 58.4% versus 50.4%. Adding the judge narrows the point-estimate gap on Gemma and Llama and reverses it on Qwen (Table 3). These descriptive differences do not establish statistically resolved advantages.

Selected members support positive reinforcement results. For Gemma-3-4B pairs whose members have guard-usable reinforcement settings, prompting reaches 19.8% [7.5, 33.7] and the hybrid reaches 41.1% [26.5, 53.9]. The hybrid improves 9 of 10 sets. Its point advantage remains at three and four instructions, although at four instructions validity is 78.0% with prompting but 18.9% with the hybrid. Selection uses the guard only, so members need not qualify under the judge.

Full-pool losses grow with frozen layer-separated edits. The hybrid falls behind prompting as sets grow (Figure 4, a separate held-out comparison with paired intervals). At two instructions, every hybrid-minus-prompt interval contains zero. At four, the differences are -6.0 [-13.0, -0.1], -22.7 [-47.2, -0.4], and -5.0 [-14.7, 2.9] points. On Gemma, hybrid validity falls from 77.3% at two instructions to 0.2% at eight, while prompt-only validity changes only from 88.8% to 83.4%. The loss therefore includes answer damage, not just harder conjunctions.

Dose tuning recovers much of the reinforcement loss. Reinforcement depends strongly on dose calibration, but tuning does not close the installation gap. As a reference, we keep layers fixed and tune one shared dose multiplier per set on development prompts. At four instructions on Gemma-3-4B's full pool, tuning recovers 20.0 points [11.9, 29.1] over the frozen hybrid. The tuned hybrid then matches prompting instead of falling below it, and it exceeds prompting at one tested count (Table 7). The same tuning recovers 4.3 points [0.8, 9.0] on Qwen2.5-7B and 4.3 [0.7, 8.1] on Llama-3.1-8B. For layer-separated installation, it changes success by only 0.0 points [-0.2, 0.0].

Changing the composition rule also avoids substantial losses. Another tested construction avoids much of the layer-separated loss. On the same held-out sets and prompts at four instructions, fixed-norm reinforcement exceeds the layer-separated hybrid by 6.3 points [0.6, 13.2] on Qwen and 22.1 [2.7, 43.7] on Gemma. Its differences from prompting are 0.3 [-2.8, 3.3] and -0.7 [-5.4, 3.6], with both intervals containing zero. This comparison changes layer placement and norm control along with the composition rule, so it does not isolate a single cause. Gains over prompting remain limited: fixed-norm reinforcement improves Llama pairs by 5.2 points [0.4, 11.7], while every rule-versus-prompt interval contains zero at eight instructions.

Larger models retain both gains and losses. On full-pool pairs, prompting exceeds fixed-norm installation by 67.2, 61.2, and 58.0 points on Qwen2.5-32B, Gemma-3-27B, and Llama-3.1-70B, with all paired intervals excluding zero. Gemma's hybrid improves over prompting by 10.9 points [0.3, 23.7] on pairs, but the difference becomes -18.7 [-34.7, -4.6] at four instructions. At eight instructions, the hybrid trails prompting on all three models, with intervals excluding zero. Appendix D describes how these sets were selected.

Table 5: **Generic retry versus diagnostic revision on instruction pairs.** Gains over first pass in joint valid success. Additional benefit compares final outcomes (95% paired bootstrap intervals). Generations are retry / diagnostic. Both arms share the pass cap and stopping rule, on 30 sets per model. Retrying recovers much of the gain; diagnostics add a resolved gain only on Llama-3.1-8B.

Model Used for Generation	Retry gain (↑)	Diagnostic gain (↑)	Additional benefit (↑)	Generations
Qwen2.5-7B	3.9 [1.9, 6.3]	6.6 [3.7, 10.0]	2.5 [−0.1, 5.3]	1.744 / 1.703
Gemma-3-4B	4.2 [1.8, 7.5]	6.8 [2.8, 11.9]	2.3 [−0.6, 5.7]	1.828 / 1.794
Llama-3.1-8B	5.6 [3.6, 7.7]	10.7 [7.1, 14.7]	5.6 [2.9, 8.8]	1.906 / 1.848

5 Prompting and checker-guided revision

Prompting already beats installation edits, so the real bar for steering is the best cheap prompt-side method. We test whether wording, checker-guided revision, or cheap post-processing raises that bar.

Wording rarely changes success. Rewording the instruction list rarely changes joint valid success. At two instructions, emphasis, reversed order, and a development-selected template give nine pairwise contrasts against the plain list, all with intervals containing zero. At six instructions, emphasis improves Gemma-3-4B by 1.8 points [0.1, 4.0]. We therefore keep the plain list as the composition baseline throughout (Appendix H).

Generic retries recover much of the revision gain. Checker-guided revision starts from the instruction list. After a failed strict check, the next turn includes the previous answer and the missed instruction sentences, for up to 3 generations. A matched generic retry instead states only that the answer failed to follow the instructions. Both arms use the same sets, prompts, pass cap, decoding, and checker-based stopping rule; the judge scores outputs but provides no feedback. Generic retries improve joint valid success by 3.9 to 5.6 points (Table 5). Detailed diagnostics add a further 2.5, 2.3, and 5.6 points to the final outcome on Qwen, Gemma, and Llama. Only Llama's interval excludes zero. Diagnostics also use slightly fewer generations. Appendix M.1 discusses the remaining compute and output differences between the arms.

Cheap post-processing also helps where it applies. Deterministic transforms of the answers need no search and no weight change. They raise joint valid success by 3.6 [0.9, 7.5], 8.4 [2.2, 16.3], and 6.7 [2.0, 13.2] points across models, but cover only 7 of the 25 constraint types. Our reimplementation of learned steering tokens also does not consistently beat the prompt (Appendix M).

6 Related work

Activation steering and internal control. Activation steering changes internal representations at inference time without updating weights. Activation Addition, Contrastive Activation Addition, and Inference Time Intervention derive or apply behavioral directions, while stronger edits can trade target behavior against helpfulness (Turner et al., 2024; Rimsky et al., 2024; Li et al., 2023). Representation Engineering frames these methods more broadly (Zou et al., 2025). Mean centering, task vectors, and function vectors further suggest that behaviors can be encoded compactly and reused across contexts, with limited evidence of semantic composition (Jorgensen et al., 2023; Hendel et al., 2023; Todd et al., 2024). These results motivate modular control but do not show that independently calibrated instruction directions remain reliable once frozen and combined.

Calibration and answer quality. Steering depends on dose, layer, input, and target behavior. Prior work calibrates instruction steering on held-out examples, evaluates instruction-blind quality, or gates and adapts edits to preserve coherence (Stolfo et al., 2025; Subbiah et al., 2026; Chalnev et al., 2024; Lee et al., 2025; Kang & Kim, 2026). Table 24 summarizes selected studies. Dynamic composition also finds property specific settings and a strength fluency tradeoff (Scalena et al., 2024). AxBench evaluates concept expression, relevance, and fluency and finds prompting stronger than tested representation-based methods (Wu et al., 2025), while ReIFE shows that the accuracy of LLM evaluators depends on the base model and protocol (Liu et al., 2025). We therefore separate checker compliance from instruction-blind quality, check the judge against a human audit, and freeze settings before judging so quality never triggers reselection.

Composition and modularity. Task arithmetic and function vectors show that learned behaviors can sometimes combine in parameter or activation space (Ilharco et al., 2023; Todd et al., 2024). Activation steering work further composes directions through summation, separate intervention sites,

dynamic strength control, steering tokens, or training free attribute combinations (van der Weij et al., 2024; Scalena et al., 2024; Subbiah et al., 2026; Radevski et al., 2026; Kang et al., 2026). These studies establish that composition is possible but, to our knowledge, leave open whether directions calibrated independently remain reliable once frozen and reused on unseen instruction sets. We test this setting directly, separating failures due to limited individual coverage from those introduced by composition and calibration. We compare several composition rules and per-set tuning. We include a geometry-aware correction (Pan et al., 2026), damage controls (Luo et al., 2026), and baseline comparisons (Pres et al., 2024) against matched prompting and retry. Malla et al. (2026) instead trace side effects to model defaults, not to direction geometry.

Constraint benchmarks and inference-time alternatives. IFEval and IFBench provide verifiable constraints (Zhou et al., 2023; Pyatkin et al., 2025). FollowBench, Multi-IF, InFoBench, and ComplexBench extend evaluation to harder, multi-turn, decomposed, and interacting instructions (Jiang et al., 2024; He et al., 2024; Qin et al., 2024; Wen et al., 2024). These benchmarks measure compliance, but checker success need not imply a useful answer (Section 3). Self-Refine, Reflexion, and CRITIC improve outputs through repeated generation and feedback (Madaan et al., 2023; Shinn et al., 2023; Gou et al., 2024). We therefore compare prompting with generic retries and checker-guided revision to separate gains from another attempt from gains due to detailed diagnostics. Because checker feedback follows benchmark rules, it can inherit their gap between checker success and useful answers. Constrained decoding and weight editing offer other routes to control (Willard & Louf, 2023; Perin et al., 2026).

7 Discussion and Limitations

The evidence supports conditional, not universal, composition. Our experiments cover one extraction rule, finite calibration searches, greedy decoding, and open-ended writing tasks with deterministic constraints. The successful installation study on Gemma-3-27B further covers one model and a small selected population. Dose tuning and alternative composition rules recover substantial reinforcement losses, showing that some failures come from calibration and intervention design rather than composition alone. They do not, however, provide a general recipe for composition or establish an impossibility result. The larger models also use fewer instances and prompts, and witness screening becomes increasingly selective, retaining only 1 set at fifteen instructions. Our conclusions therefore support composition under identifiable conditions rather than uniform reliability across models, instructions, and set sizes.

The judge measures more than relevance, but imperfectly. We audit 200 guard-valid answers from the prompt-only and hybrid arms of the three smaller composition models. Half were accepted by the judge and half were rejected. 3 human raters assess whether each answer responds to the task, contains a noticed factual error, or has broken writing. Majority agreement with the judge reaches 80.0% ($\kappa = 0.60$) when acceptance requires a full response, no noticed error, and intact writing. Agreement falls to 61.0% ($\kappa = 0.22$) for full task response alone and 51.0% ($\kappa = 0.02$) when partial responses count, suggesting that the judge aligns more closely with combined answer quality than relevance alone. Appendix K.2 details the blinded audit and sampling limits, while Appendix N shows judge-accepted answers that the human majority rejects or marks unsure.

8 Conclusion

We studied whether independently calibrated activation edits can carry multiple instructions while preserving answer quality. Our results show that composition is constrained first by **coverage**, since many instructions do not produce reliable individual edits, and then by **quality**, which further reduces the number of apparent successes after instruction blind evaluation. When individually successful installation directions are composed, they outperform the unedited model, although explicit prompting remains stronger. Much of the remaining gap depends on **calibration**: dose tuning and alternative composition rules recover substantial losses that appear when edit settings are kept fixed. These findings suggest that future work on composable steering should evaluate answer quality independently of instruction checkers, compare steering directly against simply stating the instructions in the prompt, and report how each edit is calibrated. More broadly, the results imply that the practical value of composable steering depends less on whether multiple edits can be combined at all and more on whether they remain competitive with prompting under realistic quality and calibration constraints. Composable steering is feasible, but it still has to earn its place against prompting.

AI USE STATEMENT

Generative AI tools assisted with manuscript drafting, literature review, interpretation of reported results, and editorial verification. Claude assisted with earlier drafts; OpenAI Codex assisted with the present revision, including protocol checks and manuscript restructuring. Language models also generated witness answers and supplied the answer-quality judgments described in Appendices D and K. The authors reviewed the assisted work and are responsible for the final text, claims, and artifacts.

ETHICS STATEMENT

The experiments use public instruction-following benchmarks and human-written task collections. Three authors volunteered for the human annotation study and consented to use of their ratings. No external participants were recruited, and the material shown contained no personal data. The authors considered separate ethics review unnecessary for this author-only evaluation of model outputs. The interface withheld model and arm identities, target instructions, and judge verdicts; author participation remains a limitation despite these safeguards (Appendix K.2).

REPRODUCIBILITY STATEMENT

Section 2 defines the experimental conditions and outcomes. Appendices A and J specify calibration and intervention rules. The supplementary material reports the single-instruction curves and frozen settings. Anonymized code for direction extraction, calibration, composition, revision, and judging is available at `https://anonymous.4open.science/r/ISC`.

The compute ledger records 73 instances of eight H100 accelerators, totaling 1,619 allocated instance-hours or about 12,954 accelerator-hours. Recorded judge usage for the single-instruction frozen tests is 382.3 million input tokens and 26.9 million output tokens.

REFERENCES

Marah Abdin et al. Phi-3 technical report: A highly capable language model locally on your phone, 2024. URL https://arxiv.org/abs/2404.14219.

Dyah Adila, Shuai Zhang, Boran Han, and Bernie Wang. Discovering bias in latent space: An unsupervised debiasing approach. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 246–261. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/adila24a.html.

Sviatoslav Chalnev, Matthew Siu, and Arthur Conmy. Improving steering vectors by targeting sparse autoencoder features, 2024. URL https://arxiv.org/abs/2411.02193.

Databricks. Free Dolly: Introducing the world's first truly open instruction-tuned LLM. Blog post, 2023. URL https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm.

Gemma Team. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Zhibin Gou, Zhihong Shao, Yeyun Gong, yelong shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Sx038qxjek.

Aaron Grattafiori et al. The Llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Yun He, Di Jin, Chaoqi Wang, Chloe Bi, Karishma Mandyam, Hejia Zhang, Chen Zhu, Ning Li, Tengyu Xu, Hongjiang Lv, Shruti Bhosale, Chenguang Zhu, Karthik Abinav Sankararaman, Eryk

Helenowski, Melanie Kambadur, Aditya Tayade, Hao Ma, Han Fang, and Sinong Wang. Multi-IF: Benchmarking LLMs on multi-turn and multilingual instructions following, 2024. URL https://arxiv.org/abs/2410.15553.

Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9318–9333, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.624. URL https://aclanthology.org/2023.findings-emnlp.624/.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6t0Kwf8-jrj.

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

Ole Jorgensen, Dylan Cope, Nandi Schoots, and Murray Shanahan. Improving activation steering in language models with mean-centring, 2023. URL https://arxiv.org/abs/2312.03813.

Hyun Gu Kang, Daniil Gurgurov, Tanja Baeumel, Josef van Genabith, and Simon Ostermann. Compositional multilingual and behavioral attribute steering, 2026. URL https://arxiv.org/abs/2609.08410.

Minjae Kang and Jaehyung Kim. Enhancing instruction following of LLMs via activation steering with dynamic rejection. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=OpuPBNcQwe.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. OpenAssistant conversations – democratizing large language model alignment, 2023. URL https://arxiv.org/abs/2304.07327.

Bruce W. Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre Dognin, Manish Nagireddy, and Amit Dhurandhar. Programming refusal with conditional activation steering, 2025. URL https://openreview.net/forum?id=Oi47wc10sm.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 41451–41530. Curran Associates, Inc., 2023. doi: 10.52202/075280-1797. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/81b8390039b7302c909cb769f8b6cd93-Paper-Conference.pdf.

Yixin Liu, Kejian Shi, Alexander Fabbri, Yilun Zhao, PeiFeng Wang, Chien-Sheng Wu, Shafiq Joty, and Arman Cohan. ReIFE: Re-evaluating instruction-following evaluation. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 12247–12287, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.610. URL https://aclanthology.org/2025.naacl-long.610/.

Daming Luo, Christy Liang, and Junyu Xuan. SteerCheck: Attribution specificity and alignment leakage in activation-steering audits, 2026. URL https://arxiv.org/abs/2608.24335.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. SELF-REFINE: iterative refinement with self-feedback. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Srikanth Malla, Chiho Choi, and Joon Hee Choi. Steering interference reflects the model's defaults, not the behavior directions, 2026. URL https://arxiv.org/abs/2609.06951.

Weici Pan, Xander Barron, Jiawei Zhou, and Zhenhua Liu. Spillover-aware multi-value steering for pluralistic LLM alignment, 2026. URL https://arxiv.org/abs/2609.05800.

Gabriel J. Perin, Lucas Boscaini, André Araujo, and Nina S. T. Hirata. Training-free task vectors for LLM behavioral control, 2026. URL https://arxiv.org/abs/2609.09054.

Itamar Pres, Laura Ruis, Ekdeep Singh Lubana, and David Krueger. Towards reliable evaluation of behavior steering interventions in LLMs, 2024. URL https://arxiv.org/abs/2410.17245.

Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hanna Hajishirzi. Generalizing verifiable instruction following. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference. Curran Associates, Inc., 2025. doi: 10.52202/085713-1645. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/46499a0622ecf568b72d17b61e45dbd5-Paper-Datasets_and_Benchmarks_Track.pdf.

Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. InFoBench: Evaluating instruction following ability in large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 13025–13048, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.772. URL https://aclanthology.org/2024.findings-acl.772/.

Qwen. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Gorjan Radevski, Kiril Gashteovski, Giwon Hong, Carolin Lawrence, and Goran Glavaš. Compositional steering of large language models with steering tokens. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31087–31104, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1435. URL https://aclanthology.org/2026.acl-long.1435/.

Nazneen Rajani, Lewis Tunstall, Edward Beeching, Nathan Lambert, Alexander M. Rush, and Thomas Wolf. No Robots. https://huggingface.co/datasets/HuggingFaceH4/no_robots, 2023.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

Daniel Scalena, Gabriele Sarti, and Malvina Nissim. Multi-property steering of large language models with dynamic activation composition. In Yonatan Belinkov, Najoung Kim, Jaap Jumelet, Hosein Mohebbi, Aaron Mueller, and Hanjie Chen (eds.), *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 577–603, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.blackboxnlp-1.34. URL https://aclanthology.org/2024.blackboxnlp-1.34/.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Alessandro Stolfo, Vidhisha Balachandran, Safoora Yousefi, Eric Horvitz, and Besmira Nushi. Improving instruction-following in language models through activation steering. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=wozhdnRCtw.

Melanie Subbiah, Zara Hall, and Kathleen McKeown. On the limits of steering vectors for preference-aligned generation, 2026. URL https://arxiv.org/abs/2607.01802.

Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. Function vectors in large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=AwyxtyMwaG.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering, 2024. URL https://arxiv.org/abs/2308.10248.

Teun van der Weij, Massimo Poesio, and Nandi Schoots. Extending activation steering to broad skills and multiple behaviours, 2024. URL https://arxiv.org/abs/2403.05767.

Bosi Wen, Pei Ke, Xiaotao Gu, Lindong Wu, Hao Huang, Jinfeng Zhou, Wenchuang Li, Binxin Hu, Wendy Gao, Jiaxing Xu, Yiming Liu, Jie Tang, Hongning Wang, and Minlie Huang. Benchmarking complex instruction-following with multiple constraints composition. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=U2aVNDrZGx.

Brandon T. Willard and Rémi Louf. Efficient guided generation for large language models, 2023. URL https://arxiv.org/abs/2307.09702.

Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. AxBench: Steering LLMs? even simple baselines outperform sparse autoencoders. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 67035–67080. PMLR, 2025. URL https://arxiv.org/abs/2501.17148.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and Chatbot Arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models, 2023. URL https://arxiv.org/abs/2311.07911.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to AI transparency, 2025. URL https://arxiv.org/abs/2310.01405.

Appendix

Table of Content

A Protocol details **15**

B Models Used **17**

C Single-instruction curves **17**

D Composition members **19**

E Threshold sensitivity **22**

F Numerical repeat controls **23**

G Random-direction distributions **26**

H Prompt variants **26**

I Prediction protocol and diagnostics **28**

J Composition geometry and ridge diagnostics **29**

K Quality evaluation and human validation **31**

K.1 LLM Judge 31

K.2 Human validation 32

K.3 Additional judge diagnostics 33

L Matched decomposition of composition loss **33**

L.1 Extending the decomposition to already-satisfied instructions 34

M Practical baselines and controls **35**

M.1 Revision and matched retry 37

N Checker success does not guarantee answer quality **38**

O Official IFBench evaluation **39**

P Rubric sensitivity **40**

Q Predicting unseen combinations **41**

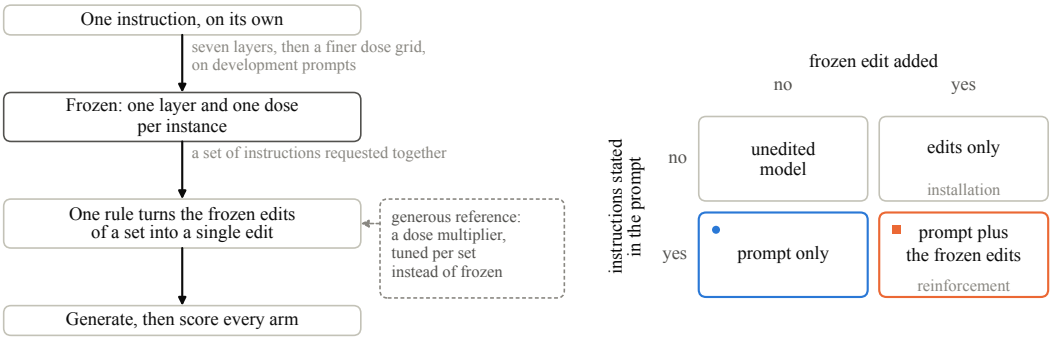


Figure 5: **Calibration and the comparison arms.** For each instruction instance and prompt condition, development prompts select one layer and one dose. Shared-layer rules derive a common layer and dose from those settings. Layer-separated edits keep the individual settings. The hybrid uses layer-separated reinforcement edits. The separate reference tunes a dose multiplier for each set. For corrected rules, it also tunes a ridge.

A PROTOCOL DETAILS

Calibration. Figure 5 summarizes calibration and the comparison arms. We use only the automatic guard during calibration and fix the direction seed to 0. For each constraint and prompt condition, the coarse search tests instance 0 on 40 development prompts. It tries seven layers and the doses $\{0.05, 0.10, 0.20, 0.40, 0.80, 1.20\}$. We spread the candidate layers across each model: 4, 8, 12, 16, 20, 24, 27 for Qwen2.5-7B, 5, 10, 15, 19, 24, 29, 33 for Gemma-3-4B, 5, 9, 14, 18, 23, 27, 31 for Llama-3.1-8B and Phi-3-mini-4k-Instruct, 9, 18, 27, 37, 46, 55, 62 for Qwen2.5-32B, 9, 18, 27, 35, 44, 53, 60 for Gemma-3-27B, and 11, 23, 34, 46, 57, 69, 77 for Llama-3.1-70B. We select the layer with the most usable coarse doses, breaking ties toward the shallower layer. If no layer has a usable dose, we choose the layer and dose with the highest valid compliance among the settings that satisfy the quality bounds. Ties between layers follow the same rule.

Fine grid and frozen dose. We build each instance's fine grid from the coarse results at the selected layer. If any coarse dose is usable, we add ten interior doses with geometric spacing. The lower end is the largest nonusable coarse dose below the first usable dose. The upper end is the first higher coarse dose that violates a quality bound. A missing lower end becomes 0.025, and a missing upper end becomes 1.60. The six coarse doses remain in the grid. If no coarse dose is usable, we use $0.05 \cdot 2^{k/3}$ for $k = 0, \ldots, 15$, together with 1.20 and the original coarse doses. We round doses to four decimal places and remove duplicates. Small models evaluate all three instances on 80 development prompts, while larger models evaluate instance 0 on 40. For each instance, we find the longest run of consecutive usable doses and freeze the dose in its middle. If the run has even length, we take the lower of the two middle doses. If several runs tie, we take the run with the lower doses. If no dose is usable, we select the dose with the highest valid compliance among the doses that pass quality. Ties go to the lower dose. If no dose passes quality, we choose the lowest dose in the grid.

Need, onset, and installation claims. We define *need* as the lowest usable dose on the fine development grid. We define *onset* as the first higher tested dose that violates either quality bound. Need is undefined when no tested dose is usable. Onset is undefined when need is undefined, or when no higher tested dose violates quality. On the frozen test set, we claim an installation only when three conditions hold: (i) unedited valid compliance is below 0.80, (ii) the edit raises valid compliance by at least 20 percentage points, and (iii) the lower end of a paired 95% bootstrap interval is above zero. The interval uses 10,000 prompt resamples with replacement and seed 0. The edited and unedited arms share the same resampled prompts. The analysis with both the guard and the judge applies the same criteria to the same frozen settings, but with the full validity rule.

Composition uncertainty. We compute composition intervals from 10,000 bootstrap draws with seed 0 and report the 2.5th and 97.5th percentiles. For fully judged families, each draw resamples sets and prompts with replacement and shares the prompt sample across all sets and arms. For families judged on a fixed subset, each draw resamples sets and then resamples the judged positions within each set. If a set is drawn more than once, all copies reuse the same position sample. Arms share the same resamples, and paired contrasts use per answer differences. Comparisons between the

Table 6: Every composition comparison reported in the paper. The full pool draws from all twenty-five instructions. The members that work alone are those with a usable setting on that model. Judged prompts are the positions with a judge verdict. They are fixed per set before judging, so every arm of a comparison is scored on the same positions. Sets per K varies with how many sets of that size passed the witness screen.

Comparison	Models	K	Sets	Judged	Settings
Deployment, full pool	3	1, 2, 3, 4, 6, 8, 12, 15	1 to 75	200	frozen
Deployment, works-alone pool	3	1, 2, 3, 4	5 to 11	200	frozen
Generous reference	3	2, 3, 4, 6, 8, 12, 15	1 to 30	100	tuned per set
Central comparison	3	2, 4, 8	10	200	frozen
Larger models	3	2, 4, 8	20	100	frozen

generous reference and deployment use the reference's judged positions for both arms. Answers that the guard marks invalid count as failures at their predefined positions. They stay in the denominator. Intervals are unadjusted 95% intervals. Our main claims rest on effects that repeat across models rather than on any single interval.

Prompt conditions. Every example contains a single user message. Without instructions, the message contains only the base task. With instructions, we append the instruction sentences in set order, separated by single spaces, following IFEval prompt construction. Each constraint sentence is the corresponding IFEval description with its instance arguments filled in. The prompt variants in Section 5 modify only this stated instruction block. *Reverse* lists the instructions in reverse order. *Structured* writes the task, a blank line, a header, and a numbered list with one instruction per line. The header is "Follow all of these requirements in your response:" or "Your response must obey every rule below." *Priority* keeps the full instruction list. After it, *Priority* adds "Above all, make sure you satisfy these:" and then the boosted instructions.

Chat template. We apply each model's own chat template with one user turn and the generation prompt. We add no system message, so templates retain any default text they normally insert. We tokenize the rendered prompt without adding extra special tokens. If the tokenizer uses a beginning of sequence token, we keep exactly one.

Decoding. We use greedy decoding throughout. We left pad batches so that the final position in every row contains a real token. For each model and response language instance, we choose a limit on new tokens. It is the smallest of 1,536, 2,048, 3,072, and 4,096 at which at most 3% of unedited answers are cut off. If no limit meets this, we use 4,096. Cells containing multiple instructions use the largest limit among their members. From fine grid evaluation onward, we stop a row early if its last 128 generated tokens repeat exactly with a period of at most 32 tokens. We check this condition every 16 tokens after the row reaches 256 new tokens, and we count such outputs as cut off. We batch the unedited repeat arm only with itself. Its difference from the unedited arm therefore measures the numerical variation between runs.

Intervention rule. We add each edit to the output of decoder layer ℓ, which is the residual stream after that layer. We add it at every prompt position and every generated position. For a relative edit e, each row receives $\|h_\ell\|\, e$ at every position in a forward call, where h_ℓ is that row's residual at its newest position. We recompute this reference norm at every decoding step and separately for each row, so one row's edit never depends on other rows in the batch. A single direction uses $e = s\, d/\|d\|$ for dose s. Composed edits follow Appendix J, and normalized rules use Equation 4. The hook does not renormalize e. The coefficient of Stolfo et al. (2025) instead depends directly on the activation. For format instructions, each position receives $(\bar{z} - h \cdot u)\, u$. Here u is a unit direction, and $\bar{z}$ is the mean projection from prompts that state the instruction. We multiply this term by a gain γ, so $\gamma = 1$ recovers their rule. This rule does not scale by the residual norm.

Reported numbers. We round percentages and percentage point differences half away from zero to the number of decimals shown.

Comparison map. Table 6 lists every composition comparison. For each one, it gives the population, instruction count, sets, judged positions, and frozen or tuned settings. This makes the conditions behind each reported result explicit.

Selection always returns a positive dose. Our selection procedure always returns an edit. If no candidate dose satisfies the quality bounds, we still choose the best available dose instead of no edit.

Table 7: Per-set dose tuning of the reinforcement hybrid on the Gemma-3-4B full pool, paired on the reference's sets and judged positions. The middle column is the gain over the same rule at its frozen setting. The right column compares the tuned arm with stating the instructions. Positive favours the tuned hybrid. Only the six-instruction contrast against the prompt excludes zero.

Instructions	Tuned minus frozen	Tuned minus prompt
Two	2.8 [−0.2, 6.0]	6.0 [−1.2, 13.8]
Three	9.3 [4.6, 14.4]	3.1 [−1.5, 7.8]
Four	20.0 [11.9, 29.1]	4.8 [−0.7, 10.2]
Six	7.3 [2.8, 12.7]	4.1 [0.8, 7.9]
Eight	1.3 [0.2, 2.9]	−3.0 [−8.6, 0.5]

As a result, every frozen setting has a positive dose. In contrast, Stolfo et al. (2025) compare validation performance with and without steering and apply no edit at test time when no layer improves. Our coverage results therefore characterize a mandatory edit protocol.

B Models Used

Table 8 lists the public models we use.

Table 8: Public models used in our experiments.

Model
Qwen2.5-32B-Instruct
Qwen2.5-7B-Instruct
gemma-3-27b-it
gemma-3-4b-it
Llama-3.1-70B-Instruct
Llama-3.1-8B-Instruct
Phi-3-mini-4k-instruct

C Single-instruction curves

Guard-only outcome cases. Table 9 gives the complete single-instruction outcomes under the automatic guard. This guard is the check used during selection. Appendix E gives the same classifications based on the judge, and at alternative targets. Appendix A defines the calibration rules. Figure 6 shows selected dose curves.

Table 9: **Single-instruction outcomes on the frozen test.** Counts are over constraint instances, with n per prompt condition. The larger models use one instance. S: already satisfied; U: usable setting found; I: improvement only in unacceptable outputs; N: no adequate improvement found in the tested search; Claim: installation claims. Validity uses the automatic guard. Without the instruction in the prompt, most instances fall in N, and a usable setting is rare. With the instruction in the prompt, the unedited model already satisfies most of them, except on Phi-3-mini (29 of 75).

		Instruction not in the prompt					Instruction in the prompt			
Model	n	S	U	I	N	Claim	S	U	I	N
Qwen2.5-7B	75	9	0	7	59	7	51	5	2	17
Gemma-3-4B	75	12	0	5	58	5	49	9	4	13
Llama-3.1-8B	75	9	4	1	61	5	40	11	7	17
Phi-3-mini	75	8	0	6	61	5	29	10	5	31
Qwen2.5-32B	25	4	2	3	16	5	20	2	1	2
Gemma-3-27B	25	0	6	2	17	8	21	3	0	1
Llama-3.1-70B	25	4	3	3	15	6	20	2	1	2

The supplementary material contains the complete development-dose curves (Supplementary Figures 1 to 56) and the frozen calibration banks (Supplementary Tables 1 to 7).

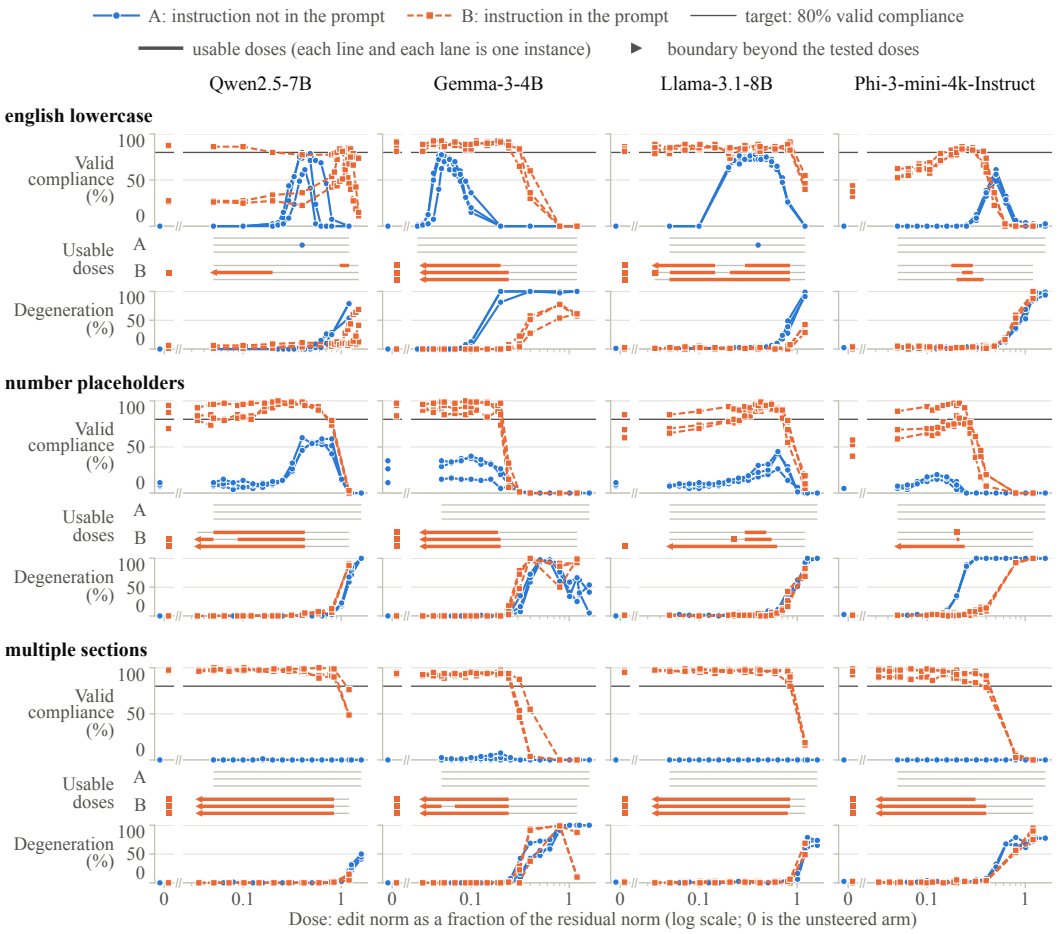


Figure 6: **Selected single-instruction operating regions on development prompts.** Validity uses the automatic guard alone. Columns show models. Each line and each usable-dose lane is one instruction instance. Blue circles mean the instruction is absent from the prompt. Orange squares mean the instruction is stated in the prompt. A zero-dose marker means the unedited arm is already usable. An arrowhead means the boundary lies beyond the tested doses. Each condition uses its own selected layer. The first two constraints have usable doses in the most pairs of model and condition. The last constraint has the strongest prompted contrast among the constraints that have no usable or already satisfied installation. The supplementary material gives the complete curves.

Table 10: Paired differences in joint valid success for the successful-installation test on Gemma-3-27B. Values are points with 95% bootstrap intervals, on the first 100 test positions and on all 400.

	Two instructions		Three instructions	
Contrast	First 100	All 400	First 100	All 400
Layer-separated − unedited	19.8 [−8.2, 49.1]	35.0 [13.7, 57.8]	5.9 [−27.1, 37.5]	17.2 [−8.5, 43.0]
Fixed-norm − unedited	3.2 [−5.7, 18.2]	13.4 [5.2, 25.0]	11.9 [−2.1, 30.9]	16.9 [4.6, 32.9]
Orthogonalized − unedited	2.5 [−11.7, 23.4]	12.0 [1.1, 28.8]	−3.0 [−26.8, 22.3]	2.8 [−15.9, 23.8]
Prompt only − layer-separated	18.6 [11.0, 26.6]	17.2 [11.2, 23.5]	52.4 [39.6, 65.7]	49.0 [36.5, 61.7]

D COMPOSITION MEMBERS

Instructions and instances. We represent each of the 25 IFEval instructions with three instances (listed in Supplementary Table 8). Each instance fixes the instruction arguments with IFEval's argument generator and fixed seeds. It also records the exact sentence shown to the model. When an instruction type provides fewer than three argument settings, we complete the set with paraphrases of its first instruction sentence.

Set construction and admission. We write $S_{K,j}$ for full-pool set j of size K and $U_{K,j}$ for a set drawn from the guard-qualified population. The full pool contains compatible sets with $K \in \{2, 3, 4, 6, 8, 12, 15\}$. It has up to 30 sets per size, and the instances within each set are fixed. We admit a set only when it has a passing witness answer on each of three designated development prompts. A witness answer passes when it satisfies every member checker and the quality judge. This screen evaluates the set itself. It is separate from the usable-setting rule in Section 2, which evaluates an edit dose. We build each witness answer in one of two ways: from the human reference by a rule, or with a model in up to five checker-guided attempts. If a set fails, we replace it with another seeded draw. For $K = 15$, we may instead redraw its instances up to three times. Supplementary Table 9 lists all admitted full-pool sets. Some populations use only members that already work individually. For these, we sample sets with the same procedure from each model's usable instruction pool.

What the witness screen discarded. Table 11 gives the screen's counts per instruction count. At two instructions it discarded nothing: 0 of the 30 sets drawn failed it. Thus, no pair was removed by the screen before the two-instruction comparisons in Section 4. The screen grows stronger as the count rises, because one answer has to satisfy every checker at once. It discarded 3 of 33 sets at three instructions and 157 of 189 at twelve. At fifteen instructions it left 1 admitted set out of 39 drawn. This is a coverage limit at that count, not a result about steering.

Larger-model sets are a selected population. The full-pool sets used for the larger models differ from the successful-installation population of Table 4. Their witness construction also uses a transfer model. This makes them a selected population.

Two readouts of the successful-installation test. The first readout of Table 4 scores test positions 0 to 99 of each set. Its paired intervals were too wide to resolve a gain over the unedited model. We therefore extended the same sets, arms, and frozen settings to positions 100 to 399. The main text reports all 400 positions. Table 10 gives both readouts. Prompting exceeds layer-separated composition in both. The gains over the unedited model are larger on all 400 positions and are resolved only there.

Part of the gain reflects answer validity. On all 400 positions at two instructions, 74.1% of unedited answers are valid, compared with 86.4% for layer-separated composition and 96.3% for prompting. The unedited model never degenerates (0.0%). Most of its invalid answers are cut off at the token limit. These cut-off answers are 23.4% of all its answers. Among valid answers, the share that satisfies both instructions still rises from 57.8% for the unedited model to 90.1% for layer-separated composition. The gain over the unedited model is therefore not only a validity effect.

Full instance and set lists. Supplementary Tables 8 and 9 record the exact instruction instances and composition sets used in our experiments. It lists all three instances of each instruction, with the filled IFEval sentence shown to the model. It also lists every full-pool set that passes the witness screen.

Table 11: The witness screen of the full instruction pool, per instruction count K. A set is kept only if all three of its witness answers pass every member checker and the judge. Failed sets are redrawn. Passed, not needed: the set passed after the count already had enough admitted sets. At $K = 2$, the screen discarded nothing.

K	Screened	Kept	Discarded	Passed, not needed	Discarded (%)
2	30	30	0	0	0.0
3	33	30	3	0	9.1
4	39	30	9	0	23.1
6	47	30	17	0	36.2
8	80	30	49	1	61.3
12	189	30	157	2	83.1
15	39	1	38	0	97.4

Every rule at every instruction count. Table 12 reports the main deployment comparison at one and two instructions. Supplementary Table 10 extends this analysis to every instruction count, with one row for each composition rule and count.

Full composition grid. In Supplementary Table 10, the fixed-vector rules remain at or near zero beyond two instructions. The pooled results summarize performance across sets. Supplementary Table 11 shows how much single sets vary.

Per-set results. Supplementary Table 11 reports one row for every two-instruction set. For each set, it gives joint valid success for prompting alone and for prompting with frozen edits. It also gives their difference in percentage points and each member's valid pass rate, in set order. The table covers three populations: the full pool at the deployment setting on all three composition models, the individually successful population on Gemma-3-4B, and the central comparison on all three models. Joint valid success requires every member checker, the guard, and the judge to pass. Member-level rates use the same validity rule. The corresponding pooled means and intervals appear in Tables 12 and 25.

On the full pool of Gemma-3-4B, at the deployment setting, adding the edits raises 14 of the 30 sets, lowers 15, and leaves 1 unchanged. The per-set change there runs from -30.0 points to 55.5 points. The two pooled means are 50.2% and 52.8%. The other two composition models also show a mix of gains and losses on the full pool. On Qwen2.5-7B the edits raise 15 of the 30 sets and lower 13. The per-set change runs from -17.5 points to 20.0 points. On Llama-3.1-8B they raise 19 of the 30 sets and lower 9. The per-set change runs from -10.5 points to 44.0 points. On the sets whose members each work alone with the instruction in the prompt, the edits raise 9 of the 10 sets and lower 1. The per-set change runs from -2.5 points to 45.0 points. On Gemma-3-4B, the central comparison shows the same pattern on its own prompts. The pooled change is 1.6 points [-13.3, 18.8]. Yet 3 of 10 sets rise, 7 fall, and the per-set change runs from -28.0 points to 53.5 points. A pooled gain of a few points therefore says little about what one set will do. The table and these counts cover sets of two instructions. For larger sets, we report only pooled results, in the scaling analysis of Section 4.

Table 12: **Composition on the full instruction pool, at the deployment setting.** Joint valid success (percent, 95% interval over sets and prompts) on test prompts. The deployment setting takes every layer, dose, and gain from single-instruction calibration, plus one ridge per model. $K = 1$ is the single-instruction reference, where all fixed-vector rules apply the same edit. The last three columns are at $K = 2$: a single member's average pass rate, validity under the guard and the judge, and the degeneration rate. At two instructions, every fixed-vector interval contains the unedited model, and prompting beats all of them. Several rules also lower validity or raise degeneration.

Arm	Joint valid success		At $K = 2$		
	$K = 1$	$K = 2$	Mean member	Validity	Degenerate
Qwen2.5-7B					
Sets	75	30			
Unedited	14.6 [8.2, 21.7]	4.0 [0.0, 10.6]	16.6	87.6	0.0
Fixed-norm addition	18.8 [12.0, 26.1]	4.0 [0.0, 10.6]	16.6	86.1	0.1
Calibrated sum	18.8 [12.0, 26.1]	4.2 [0.2, 10.8]	16.5	80.2	1.5
Corrected (inverse Gram)	18.8 [12.0, 26.1]	4.0 [0.0, 10.7]	16.5	85.9	0.2
Corrected, primary weighted	18.8 [12.0, 26.1]	4.0 [0.2, 10.6]	17.1	85.6	0.2
Orthogonalized	18.8 [12.0, 26.1]	4.1 [0.0, 10.6]	16.5	86.5	0.1
Orthogonalized, reverse order	–	4.1 [0.1, 10.6]	16.8	86.3	0.2
Layer-separated	18.8 [12.0, 26.1]	5.4 [0.6, 12.1]	19.6	79.5	1.2
Input-dependent coefficient	17.5 [10.9, 24.6]	3.9 [0.1, 10.1]	17.0	81.1	1.3
Prompt only	64.8 [58.3, 71.1]	45.9 [35.8, 56.2]	59.4	74.6	3.0
Prompt plus the edits (hybrid)	66.5 [60.1, 72.6]	44.9 [34.6, 55.4]	55.2	66.6	5.3
Gemma-3-4B					
Sets	75	30			
Unedited	15.4 [8.8, 22.8]	4.3 [0.0, 10.9]	18.5	87.4	0.0
Fixed-norm addition	19.3 [12.4, 26.9]	3.9 [0.0, 10.2]	18.2	87.9	0.0
Calibrated sum	19.3 [12.4, 26.9]	3.1 [0.0, 8.3]	15.8	78.3	0.4
Corrected (inverse Gram)	19.3 [12.4, 26.9]	3.8 [0.0, 9.9]	18.2	87.3	0.0
Corrected, primary weighted	19.3 [12.4, 26.9]	3.4 [0.0, 9.1]	13.2	67.8	11.9
Orthogonalized	19.3 [12.4, 26.9]	3.7 [0.0, 10.2]	13.9	72.9	6.1
Orthogonalized, reverse order	–	3.5 [0.0, 9.6]	14.4	74.0	6.7
Layer-separated	19.3 [12.4, 26.9]	4.4 [0.5, 10.3]	19.8	82.7	0.1
Input-dependent coefficient	4.3 [2.1, 7.0]	0.1 [0.0, 0.3]	0.4	2.3	54.0
Prompt only	67.7 [60.5, 74.5]	50.2 [37.2, 63.1]	66.9	86.3	0.1
Prompt plus the edits (hybrid)	71.8 [65.6, 77.4]	52.8 [41.5, 64.0]	62.7	74.4	1.5
Llama-3.1-8B					
Sets	75	30			
Unedited	15.2 [8.6, 22.6]	4.1 [0.0, 10.8]	17.1	91.3	0.4
Fixed-norm addition	19.8 [12.5, 27.6]	4.4 [0.3, 11.2]	18.3	90.7	0.5
Calibrated sum	19.8 [12.5, 27.6]	4.5 [0.4, 11.3]	18.0	88.9	1.2
Corrected (inverse Gram)	19.8 [12.5, 27.6]	4.5 [0.3, 11.4]	18.4	91.1	0.4
Corrected, primary weighted	19.8 [12.5, 27.6]	4.3 [0.3, 10.9]	18.2	91.0	0.5
Orthogonalized	19.8 [12.5, 27.6]	4.5 [0.4, 11.4]	18.5	91.1	0.5
Orthogonalized, reverse order	–	4.5 [0.3, 11.3]	18.3	91.1	0.6
Layer-separated	19.8 [12.5, 27.6]	4.9 [0.7, 11.7]	19.8	88.3	1.1
Input-dependent coefficient	19.1 [11.9, 26.8]	5.0 [0.7, 11.8]	19.4	87.3	2.0
Prompt only	63.9 [56.7, 70.7]	45.0 [33.0, 56.7]	62.4	85.3	2.2
Prompt plus the edits (hybrid)	69.4 [62.9, 75.4]	49.7 [38.8, 60.2]	63.6	79.7	3.1

E THRESHOLD SENSITIVITY

Section 2 defines usability at a valid-compliance target of 0.80. Here we repeat the headline classifications at targets of 0.50, 0.70, 0.80, and 0.90. We use both validity rules: the automatic guard alone, and the guard plus the judge from Appendix K. We keep every steering setting frozen at the 0.80 target and change only the classification threshold. Table 13 reports the four single-instruction outcome classes under the guard. Table 14 repeats the analysis with the judge. Each table gives one row per model and target, with separate blocks for installation and reinforcement.

Table 13: Single-instruction outcomes under other usable targets, counted over constraint instances. The left block (installation) is without the instruction in the prompt, and the right block (reinforcement) is with it. Settings stay frozen at the 0.80 target. Only the classification changes. Validity uses the automatic guard.

Model	Target	S	U	I	N	S	U	I	N
Qwen2.5-7B	0.5	13	5	2	55	63	1	2	9
Qwen2.5-7B	0.7	12	2	5	56	59	3	2	11
Qwen2.5-7B	0.8	9	0	7	59	51	5	2	17
Qwen2.5-7B	0.9	8	0	7	60	40	6	4	25
Gemma-3-4B	0.5	12	3	2	58	63	6	0	6
Gemma-3-4B	0.7	12	3	2	58	53	10	2	10
Gemma-3-4B	0.8	12	0	5	58	49	9	4	13
Gemma-3-4B	0.9	9	0	5	61	38	3	10	24
Llama-3.1-8B	0.5	13	4	1	57	61	2	4	8
Llama-3.1-8B	0.7	12	4	1	58	51	9	5	10
Llama-3.1-8B	0.8	9	4	1	61	40	11	7	17
Llama-3.1-8B	0.9	9	1	4	61	28	5	8	34
Phi-3-mini	0.5	10	1	5	59	45	7	3	20
Phi-3-mini	0.7	8	0	6	61	37	7	4	27
Phi-3-mini	0.8	8	0	6	61	29	10	5	31
Phi-3-mini	0.9	7	0	6	62	16	5	9	45
Qwen2.5-32B	0.5	5	3	2	15	23	1	1	0
Qwen2.5-32B	0.7	4	2	3	16	22	2	1	0
Qwen2.5-32B	0.8	4	2	3	16	20	2	1	2
Qwen2.5-32B	0.9	3	2	3	17	19	1	2	3
Gemma-3-27B	0.5	5	1	2	17	23	2	0	0
Gemma-3-27B	0.7	0	6	2	17	23	2	0	0
Gemma-3-27B	0.8	0	6	2	17	21	3	0	1
Gemma-3-27B	0.9	0	4	3	18	15	4	1	5
Llama-3.1-70B	0.5	5	4	2	14	22	1	0	2
Llama-3.1-70B	0.7	5	3	3	14	21	2	0	2
Llama-3.1-70B	0.8	4	3	3	15	20	2	1	2
Llama-3.1-70B	0.9	3	2	4	16	16	3	1	5

Table 14: Single-instruction outcomes under other usable targets, counted over constraint instances. The left block (installation) is without the instruction in the prompt, and the right block (reinforcement) is with it. Settings stay frozen at the 0.80 target. Only the classification changes. Validity requires the automatic guard and a yes verdict from the judge (Appendix K).

Model	Target	S	U	I	N	S	U	I	N
Qwen2.5-7B	0.5	13	0	7	55	60	1	3	11
Qwen2.5-7B	0.7	9	0	7	59	45	1	5	24
Qwen2.5-7B	0.8	8	0	7	60	26	5	5	39
Qwen2.5-7B	0.9	0	0	7	68	0	1	5	69
Gemma-3-4B	0.5	12	2	3	58	59	3	6	7
Gemma-3-4B	0.7	12	0	5	58	48	6	8	13
Gemma-3-4B	0.8	11	0	5	59	37	5	9	24
Gemma-3-4B	0.9	0	0	5	70	6	2	10	57
Llama-3.1-8B	0.5	13	2	3	57	56	3	6	10
Llama-3.1-8B	0.7	10	2	3	60	41	6	8	20
Llama-3.1-8B	0.8	9	0	5	61	29	5	8	33

Model	Target	S	U	I	N	S	U	I	N
Llama-3.1-8B	0.9	4	0	5	66	11	7	9	48
Phi-3-mini	0.5	10	0	6	59	40	5	6	24
Phi-3-mini	0.7	7	0	6	62	19	4	10	42
Phi-3-mini	0.8	7	0	6	62	4	3	12	56
Phi-3-mini	0.9	0	0	6	69	0	0	12	63
Qwen2.5-32B	0.5	5	0	5	15	22	0	2	1
Qwen2.5-32B	0.7	4	0	5	16	20	0	2	3
Qwen2.5-32B	0.8	4	0	5	16	18	0	2	5
Qwen2.5-32B	0.9	2	0	5	18	12	0	2	11
Gemma-3-27B	0.5	5	1	2	17	23	1	1	0
Gemma-3-27B	0.7	0	5	2	18	23	1	1	0
Gemma-3-27B	0.8	0	5	2	18	20	1	1	3
Gemma-3-27B	0.9	0	2	3	20	13	2	2	8
Llama-3.1-70B	0.5	5	0	6	14	21	0	2	2
Llama-3.1-70B	0.7	5	0	6	14	20	0	2	3
Llama-3.1-70B	0.8	4	0	6	15	18	1	2	4
Llama-3.1-70B	0.9	3	0	6	16	16	0	2	7

Excluding an instruction that conflicts with the guard. The guard requires at least twenty words. The constrained-response instruction asks for one of three four-word replies. A literal answer therefore cannot satisfy both. Its checker can still accept the required phrase inside a longer response. As a result, checker success does not guarantee literal compliance. Removing this instruction leaves 288 of the 300 instances per condition across the four smaller models. Usable installations remain at 4 under the guard and 0 with the judge. Usable reinforcements remain at 18 with the judge and fall from 35 to 31 under the guard. The judge therefore still reduces the qualifying population. After the exclusion, it removes fewer instances. For the unedited prompted model, already-satisfied instances remain at 96 with the judge versus 166 under the guard.

F NUMERICAL REPEAT CONTROLS

We use three controls to measure how much results move when the experimental setup stays fixed. First, every frozen-test cell and every central-comparison cell runs the unedited arm twice. One run is in the shared batch. The other run, the unedited repeat arm, is batched only with itself (Appendix A). The difference between these runs measures the numerical variation between runs. Table 15 reports these gaps for the frozen tests. Second, Table 16 reports exact repeats of the central comparison. Third, we rerun the central steering arms with directions extracted from different prompt samples. These direction seeds measure sensitivity to the extraction prompts rather than numerical noise. All three controls use the automatic guard alone. The paragraphs below give the judged direction-seed results.

Table 15: Numerical run-to-run variation in the single-instruction frozen tests. Each value is the absolute difference, in points, between the unedited arm and the unedited repeat arm, which is the same arm batched only with itself. We report it for valid compliance and for guard validity, over every frozen-test cell of each complete bank. Validity uses the automatic guard.

Model	Cells	Compliance gap, median	Max	Validity gap, median	Max
Qwen2.5-7B	150	0.0	6.5	0.0	2.0
Gemma-3-4B	150	0.5	7.0	0.5	3.5
Llama-3.1-8B	150	0.0	5.5	0.5	4.0
Phi-3-mini	150	0.5	6.0	0.5	3.5
Qwen2.5-32B	50	0.0	6.0	0.0	6.0
Gemma-3-27B	50	0.0	11.0	2.0	13.0
Llama-3.1-70B	50	0.0	6.0	0.0	3.0

Table 16: Exact repeats and direction seeds of the central comparison, on the reported sets. Each run gives joint valid success (percent) with the automatic guard on all central prompts. SD is the standard deviation across runs, in points. Hybrid is the prompt plus the edits. A repeat row is a second run of the same arm (Appendix A). Run 0 is the headline cell in both blocks. Both blocks use the guard alone, including the reported larger instruction counts.

Model	K	Control	Arm	Per run (%)	SD
Qwen2.5-7B	2	Repeats	Unedited	0.0 / 0.0 / 0.0 / 0.0	0.01
Qwen2.5-7B	2	Repeats	Unedited repeat	0.0 / 0.0 / 0.0 / 0.0	0.01
Qwen2.5-7B	2	Repeats	Prompt only	67.2 / 67.2 / 67.2 / 67.0	0.11
Qwen2.5-7B	2	Repeats	Prompt only repeat	67.3 / 67.3 / 67.2 / 67.2	0.04
Qwen2.5-7B	2	Repeats	Fixed-norm addition	0.0 / 0.0 / 0.0 / 0.0	0.01
Qwen2.5-7B	2	Repeats	Orthogonalized	0.0 / 0.0 / 0.0 / 0.0	0.01
Qwen2.5-7B	2	Repeats	Layer-separated	0.0 / 0.0 / 0.0 / 0.0	0.00
Qwen2.5-7B	2	Repeats	Hybrid	72.4 / 72.5 / 72.4 / 72.3	0.09
Qwen2.5-7B	2	Seeds	Fixed-norm addition	0.0 / 0.0 / 0.0 / 0.0 / 0.0	0.00
Qwen2.5-7B	2	Seeds	Orthogonalized	0.0 / 0.0 / 0.0 / 0.0 / 0.0	0.00
Qwen2.5-7B	2	Seeds	Layer-separated	0.0 / 0.0 / 0.0 / 0.0 / 0.0	0.01
Qwen2.5-7B	2	Seeds	Hybrid	72.4 / 72.4 / 71.6 / 70.2 / 70.9	0.97
Qwen2.5-7B	4	Repeats	Unedited	0.0 / 0.0 / 0.0 / 0.0	0.00
Qwen2.5-7B	4	Repeats	Unedited repeat	0.0 / 0.0 / 0.0 / 0.0	0.00
Qwen2.5-7B	4	Repeats	Prompt only	30.3 / 30.1 / 30.1 / 30.6	0.19
Qwen2.5-7B	4	Repeats	Prompt only repeat	30.2 / 30.1 / 30.3 / 30.0	0.15
Qwen2.5-7B	4	Repeats	Fixed-norm addition	0.0 / 0.0 / 0.0 / 0.0	0.00
Qwen2.5-7B	4	Repeats	Orthogonalized	0.0 / 0.0 / 0.0 / 0.0	0.00
Qwen2.5-7B	4	Repeats	Layer-separated	0.0 / 0.0 / 0.0 / 0.0	0.00
Qwen2.5-7B	4	Repeats	Hybrid	27.2 / 27.4 / 27.5 / 26.8	0.33
Qwen2.5-7B	4	Seeds	Fixed-norm addition	0.0 / 0.0 / 0.0 / 0.0 / 0.0	0.00
Qwen2.5-7B	4	Seeds	Orthogonalized	0.0 / 0.0 / 0.0 / 0.0 / 0.0	0.00
Qwen2.5-7B	4	Seeds	Layer-separated	0.0 / 0.0 / 0.0 / 0.0 / 0.0	0.00
Qwen2.5-7B	4	Seeds	Hybrid	27.2 / 27.6 / 26.9 / 27.6 / 27.8	0.36
Qwen2.5-7B	8	Repeats	Unedited	0.0 / 0.0 / 0.0 / 0.0	0.00
Qwen2.5-7B	8	Repeats	Unedited repeat	0.0 / 0.0 / 0.0 / 0.0	0.00
Qwen2.5-7B	8	Repeats	Prompt only	0.8 / 0.9 / 0.8 / 0.9	0.06
Qwen2.5-7B	8	Repeats	Prompt only repeat	1.0 / 0.8 / 0.8 / 0.9	0.08
Qwen2.5-7B	8	Repeats	Fixed-norm addition	0.0 / 0.0 / 0.0 / 0.0	0.00
Qwen2.5-7B	8	Repeats	Orthogonalized	0.0 / 0.0 / 0.0 / 0.0	0.00
Qwen2.5-7B	8	Repeats	Layer-separated	0.0 / 0.0 / 0.0 / 0.0	0.00
Qwen2.5-7B	8	Repeats	Hybrid	0.6 / 0.6 / 0.6 / 0.6	0.02
Qwen2.5-7B	8	Seeds	Fixed-norm addition	0.0 / 0.0 / 0.0 / 0.0 / 0.0	0.00
Qwen2.5-7B	8	Seeds	Orthogonalized	0.0 / 0.0 / 0.0 / 0.0 / 0.0	0.00
Qwen2.5-7B	8	Seeds	Layer-separated	0.0 / 0.0 / 0.0 / 0.0 / 0.0	0.00
Qwen2.5-7B	8	Seeds	Hybrid	0.6 / 0.7 / 0.3 / 0.1 / 0.1	0.28
Gemma-3-4B	2	Repeats	Unedited	0.0 / 0.0 / 0.0 / 0.0	0.01
Gemma-3-4B	2	Repeats	Unedited repeat	0.0 / 0.0 / 0.0 / 0.0	0.01
Gemma-3-4B	2	Repeats	Prompt only	64.7 / 64.6 / 64.5 / 64.4	0.14
Gemma-3-4B	2	Repeats	Prompt only repeat	64.6 / 64.8 / 64.2 / 64.7	0.28
Gemma-3-4B	2	Repeats	Fixed-norm addition	0.0 / 0.0 / 0.0 / 0.0	0.01
Gemma-3-4B	2	Repeats	Orthogonalized	0.0 / 0.0 / 0.0 / 0.0	0.01
Gemma-3-4B	2	Repeats	Layer-separated	0.4 / 0.5 / 0.4 / 0.4	0.02
Gemma-3-4B	2	Repeats	Hybrid	76.6 / 76.6 / 76.3 / 76.3	0.14
Gemma-3-4B	2	Seeds	Fixed-norm addition	0.0 / 0.0 / 0.0 / 0.0 / 0.0	0.01
Gemma-3-4B	2	Seeds	Orthogonalized	0.0 / 0.0 / 0.0 / 0.0 / 0.0	0.00
Gemma-3-4B	2	Seeds	Layer-separated	0.4 / 0.4 / 0.3 / 0.4 / 0.4	0.05
Gemma-3-4B	2	Seeds	Hybrid	76.6 / 75.5 / 75.1 / 74.0 / 74.3	1.01
Gemma-3-4B	4	Repeats	Unedited	0.0 / 0.0 / 0.0 / 0.0	0.00
Gemma-3-4B	4	Repeats	Unedited repeat	0.0 / 0.0 / 0.0 / 0.0	0.00
Gemma-3-4B	4	Repeats	Prompt only	43.2 / 43.2 / 43.4 / 43.6	0.21
Gemma-3-4B	4	Repeats	Prompt only repeat	43.9 / 43.6 / 43.3 / 43.5	0.24
Gemma-3-4B	4	Repeats	Fixed-norm addition	0.0 / 0.0 / 0.0 / 0.0	0.00
Gemma-3-4B	4	Repeats	Orthogonalized	0.0 / 0.0 / 0.0 / 0.0	0.00
Gemma-3-4B	4	Repeats	Layer-separated	0.0 / 0.0 / 0.0 / 0.0	0.00
Gemma-3-4B	4	Repeats	Hybrid	23.0 / 22.7 / 22.9 / 22.9	0.15
Gemma-3-4B	4	Seeds	Fixed-norm addition	0.0 / 0.0 / 0.0 / 0.0 / 0.0	0.00

Model	K	Control	Arm	Per run (%)	SD
Gemma-3-4B	4	Seeds	Orthogonalized	0.0 / 0.0 / 0.0 / 0.0 / 0.0	0.00
Gemma-3-4B	4	Seeds	Layer-separated	0.0 / 0.0 / 0.0 / 0.0 / 0.0	0.00
Gemma-3-4B	4	Seeds	Hybrid	23.0 / 21.9 / 22.9 / 24.0 / 24.4	0.98
Gemma-3-4B	8	Repeats	Unedited	0.0 / 0.0 / 0.0 / 0.0	0.00
Gemma-3-4B	8	Repeats	Unedited repeat	0.0 / 0.0 / 0.0 / 0.0	0.00
Gemma-3-4B	8	Repeats	Prompt only	9.5 / 9.4 / 9.6 / 9.5	0.08
Gemma-3-4B	8	Repeats	Prompt only repeat	9.4 / 9.3 / 9.5 / 9.6	0.13
Gemma-3-4B	8	Repeats	Fixed-norm addition	0.0 / 0.0 / 0.0 / 0.0	0.00
Gemma-3-4B	8	Repeats	Orthogonalized	0.0 / 0.0 / 0.0 / 0.0	0.00
Gemma-3-4B	8	Repeats	Layer-separated	0.0 / 0.0 / 0.0 / 0.0	0.00
Gemma-3-4B	8	Repeats	Hybrid	0.0 / 0.0 / 0.0 / 0.0	0.00
Gemma-3-4B	8	Seeds	Fixed-norm addition	0.0 / 0.0 / 0.0 / 0.0 / 0.0	0.00
Gemma-3-4B	8	Seeds	Orthogonalized	0.0 / 0.0 / 0.0 / 0.0 / 0.0	0.00
Gemma-3-4B	8	Seeds	Layer-separated	0.0 / 0.0 / 0.0 / 0.0 / 0.0	0.00
Gemma-3-4B	8	Seeds	Hybrid	0.0 / 0.0 / 0.0 / 0.0 / 0.0	0.00
Llama-3.1-8B	2	Repeats	Unedited	0.0 / 0.0 / 0.0 / 0.0	0.01
Llama-3.1-8B	2	Repeats	Unedited repeat	0.0 / 0.0 / 0.0 / 0.0	0.01
Llama-3.1-8B	2	Repeats	Prompt only	50.5 / 50.5 / 50.5 / 50.5	0.03
Llama-3.1-8B	2	Repeats	Prompt only repeat	50.2 / 50.0 / 50.4 / 50.1	0.19
Llama-3.1-8B	2	Repeats	Fixed-norm addition	0.1 / 0.1 / 0.1 / 0.1	0.01
Llama-3.1-8B	2	Repeats	Orthogonalized	0.1 / 0.1 / 0.1 / 0.1	0.01
Llama-3.1-8B	2	Repeats	Layer-separated	0.3 / 0.4 / 0.4 / 0.4	0.01
Llama-3.1-8B	2	Repeats	Hybrid	61.8 / 61.9 / 62.1 / 61.6	0.18
Llama-3.1-8B	2	Seeds	Fixed-norm addition	0.1 / 0.1 / 0.0 / 0.1 / 0.1	0.01
Llama-3.1-8B	2	Seeds	Orthogonalized	0.1 / 0.1 / 0.1 / 0.1 / 0.0	0.01
Llama-3.1-8B	2	Seeds	Layer-separated	0.3 / 0.4 / 0.4 / 0.4 / 0.5	0.05
Llama-3.1-8B	2	Seeds	Hybrid	61.8 / 61.7 / 63.0 / 62.6 / 62.7	0.58
Llama-3.1-8B	4	Repeats	Unedited	0.0 / 0.0 / 0.0 / 0.0	0.00
Llama-3.1-8B	4	Repeats	Unedited repeat	0.0 / 0.0 / 0.0 / 0.0	0.00
Llama-3.1-8B	4	Repeats	Prompt only	39.0 / 38.1 / 38.3 / 38.2	0.40
Llama-3.1-8B	4	Repeats	Prompt only repeat	38.2 / 38.4 / 38.5 / 38.1	0.19
Llama-3.1-8B	4	Repeats	Fixed-norm addition	0.0 / 0.0 / 0.0 / 0.0	0.00
Llama-3.1-8B	4	Repeats	Orthogonalized	0.0 / 0.0 / 0.0 / 0.0	0.00
Llama-3.1-8B	4	Repeats	Layer-separated	0.0 / 0.0 / 0.0 / 0.0	0.00
Llama-3.1-8B	4	Repeats	Hybrid	32.9 / 32.4 / 32.3 / 32.4	0.27
Llama-3.1-8B	4	Seeds	Fixed-norm addition	0.0 / 0.0 / 0.0 / 0.0 / 0.0	0.00
Llama-3.1-8B	4	Seeds	Orthogonalized	0.0 / 0.0 / 0.0 / 0.0 / 0.0	0.00
Llama-3.1-8B	4	Seeds	Layer-separated	0.0 / 0.0 / 0.0 / 0.1 / 0.0	0.03
Llama-3.1-8B	4	Seeds	Hybrid	32.9 / 31.4 / 30.5 / 29.3 / 28.4	1.77
Llama-3.1-8B	8	Repeats	Unedited	0.0 / 0.0 / 0.0 / 0.0	0.00
Llama-3.1-8B	8	Repeats	Unedited repeat	0.0 / 0.0 / 0.0 / 0.0	0.00
Llama-3.1-8B	8	Repeats	Prompt only	1.6 / 1.6 / 1.6 / 1.6	0.02
Llama-3.1-8B	8	Repeats	Prompt only repeat	1.4 / 1.4 / 1.6 / 1.4	0.07
Llama-3.1-8B	8	Repeats	Fixed-norm addition	0.0 / 0.0 / 0.0 / 0.0	0.00
Llama-3.1-8B	8	Repeats	Orthogonalized	0.0 / 0.0 / 0.0 / 0.0	0.00
Llama-3.1-8B	8	Repeats	Layer-separated	0.0 / 0.0 / 0.0 / 0.0	0.00
Llama-3.1-8B	8	Repeats	Hybrid	1.1 / 1.2 / 1.0 / 1.2	0.13
Llama-3.1-8B	8	Seeds	Fixed-norm addition	0.0 / 0.0 / 0.0 / 0.0 / 0.0	0.00
Llama-3.1-8B	8	Seeds	Orthogonalized	0.0 / 0.0 / 0.0 / 0.0 / 0.0	0.00
Llama-3.1-8B	8	Seeds	Layer-separated	0.0 / 0.0 / 0.0 / 0.0 / 0.0	0.00
Llama-3.1-8B	8	Seeds	Hybrid	1.1 / 1.1 / 0.7 / 0.5 / 0.5	0.34

Direction seeds under the full validity rule. For all 5 direction seeds, we judge the prompt-plus-edits arm of every two-instruction central cell. Each seed uses the same 10 sets and the same judged positions. This lets us compare extraction seeds on identical prompts under the guard and judge together. Across seeds, the standard deviation of joint valid success is 1.4 points on Qwen2.5-7B, 1.1 on Gemma-3-4B, and 0.7 on Llama-3.1-8B. Under the guard alone on the same positions, the corresponding values are 1.0, 0.5, and 0.9. Re-extracting the direction therefore moves the judged headline by about as much as the guard-only result. The judged spread is larger on two models and slightly smaller on the third.

The reference seed is not always typical. On Qwen2.5-7B, seed 0 gives 55.6%, and the seed range is 52.5% to 55.6%. On Gemma-3-4B, seed 0 gives 58.7%, and the range is 57.7% to 60.5%. On Llama-3.1-8B, seed 0 gives 51.2%, and the range is 51.2% to 52.6%. On Qwen2.5-7B, seed 0 lies at the top of the range. The hybrid's small two-instruction lead over prompting on that model (Figure 4) therefore comes from the most favorable of the 5 extractions. This lead is similar in size to the seed spread itself. The paired central-comparison interval also leaves this difference unresolved. On Llama-3.1-8B, seed 0 lies at the bottom of the range. There, the hybrid's lead comes from the least favorable extraction.

G RANDOM-DIRECTION DISTRIBUTIONS

This appendix reports the three component distributions used in Appendix L. They are isotropic random directions, compatible partner directions, and other real instruction directions. Only one primary passes the eligibility rule of the decomposition: capital-word frequency on Llama-3.1-8B at layer 27. We evaluate it on 200 test prompts at the three core points. Eligibility requires instance 0 to have both a usable frozen-test installation and a defined development need (Appendix L).

Component draws. At each core point, we hold the primary edit fixed and add a second component with a fixed magnitude. For the isotropic control, we draw 100 unit vectors orthogonal to the primary, with fixed seeds. We scale each vector to the same total norm as the real partner it replaces. Other real directions are 10 instruction directions sampled from instructions outside the primary and its selected partners. We project each direction orthogonal to the primary and normalize it. We take the 5 compatible partners at evenly spaced ranks of absolute cosine with the primary, from the smallest to the largest. They get the same projection and normalization. The added component has relative size 0.179, 0.170, and 0.283 at the three core points. The corresponding primary projections are 0.376, 0.388, and 0.388. The total relative norms are therefore 0.416, 0.423, and 0.480.

Cost of each component type. Every cell generates answers for 200 prompts. We estimate isotropic damage on 100 fixed positions. The standalone comparisons of real partners and of other real directions against the primary alone use all 200 prompts. At the three core points, isotropic directions change the primary's valid compliance by -4.5 [-9.0, 0.1], -2.3 [-6.7, 2.5], and -4.1 [-8.9, 1.0] points. Other real directions change it by -15.1 [-18.9, -11.2], -12.9 [-16.6, -9.1], and -28.0 [-32.0, -23.8] points. Compatible partners change it by -13.2 [-17.1, -9.4], -12.2 [-15.9, -8.5], and -23.4 [-27.4, -19.5] points. All of these comparisons use the automatic guard alone. The partner-specific term in Appendix L instead compares real partners with isotropic draws on the same 100 positions. It therefore cannot be recovered by subtracting the numbers above, because they come from different prompt sets.

Answer quality under random components. Under the automatic guard, isotropic components leave validity at 96.4%, 96.6%, and 94.6%. Degeneration reaches 3.2%, 3.2%, and 4.0%. These guard-only measurements separate two effects: losing the primary instruction, and general damage to the output.

H PROMPT VARIANTS

Prompt controls. We evaluate several user-message variants for each composition set, all constructed as in Appendix A. The controls include three prompts. The first is the task alone. The second is the task followed by every instruction in set order, which we call the list prompt. The third is a repeat run of the list prompt, batched on its own. We also test emphasis and a reversed instruction order. Emphasis appends "Above all, make sure you satisfy these:" and then repeats every instruction. Finally, we test two structured prompts. Each places the task before a numbered instruction list. The list header is either "Follow all of these requirements in your response:" or "Your response must obey every rule below." Table 17 reports joint valid success on development prompts. All three models select the numbered "Follow all of these requirements" template. The selection pools all instruction counts, so it can differ from the two-instruction results in Table 17 (Appendix M).

Prompted and injected instruction counts. On the admitted fifteen-instruction set, we fix four targets. They are the members with the most usable single-instruction installation settings, pooled

Table 17: Joint valid success (percent) of each prompt wording on the two-instruction composition sets, on development prompts 120 to 199. Admitted sets are pooled with equal weight. Validity uses the automatic guard alone, because these answers are never judged. The gap between the repeat run and the instruction list measures the numerical variation between runs. Table 28 gives the test-prompt comparison.

Prompt	Qwen2.5-7B	Gemma-3-4B	Llama-3.1-8B
Task alone	4.5	4.5	4.3
Instruction list	58.9	56.5	50.5
Instruction list, repeat run	59.7	57.0	49.9
Instruction list, every instruction emphasized	62.8	60.5	53.2
Instruction list in reverse order	55.0	56.8	50.3
Numbered list, "Follow all of these requirements"	62.8	56.5	54.2
Numbered list, "Your response must obey every rule"	62.2	59.5	55.0

over the three smaller composition models. We vary the number of prompted instructions P and of injected instructions I independently over $\{0, 4, 8, 15\}$. This gives sixteen arms. When $P > 0$, the prompt states the four targets and then the first $P - 4$ remaining members in set order. When $P = 0$, it states no instruction. When $I > 0$, the edit injects the four targets and then the first $I - 4$ remaining members. Each member uses its frozen installation layer and dose, and edits that share a layer are summed. When $I = 0$, we apply no edit. We score only the four targets.

Priority partitions. We define a fixed priority pool of eight instructions, listed below. Each tested partition splits this pool into two groups of four. We run each split in both orientations (Table 18). For each orientation, we compare three conditions. The first is the list prompt alone. The second is the list prompt followed by "Above all, make sure you satisfy these:" and the four boosted instructions. The third is the list prompt with an orthogonalized edit over all fifteen directions. The edit uses the most common frozen reinforcement layer. It places the boosted members first and weights them three times more than the others. It then scales the result to the mean frozen reinforcement dose. A second pool repeats this comparison on the central prompts (Appendix M). It has six instructions from the same set, all outside the priority pool.

- A: `detectable_content:number_placeholders`
- B: `detectable_content:postscript`
- C: `detectable_format:number_bullet_lists`
- D: `detectable_format:number_highlighted_sections`
- E: `keywords:existence`
- F: `length_constraints:number_sentences`
- G: `punctuation:no_comma`
- H: `startend:end_checker`

Table 18: The 35 splits of the fixed priority pool into two halves of four, on set $S_{15,1}$, with the instructions lettered as in the list above. Each split runs twice: orientation a boosts the first half, and orientation b boosts the second.

Split	Boosted in orientation a	Boosted in orientation b
00	A, B, C, D	E, F, G, H
01	A, B, C, E	D, F, G, H
02	A, B, C, F	D, E, G, H
03	A, B, C, G	D, E, F, H
04	A, B, C, H	D, E, F, G
05	A, B, D, E	C, F, G, H
06	A, B, D, F	C, E, G, H
07	A, B, D, G	C, E, F, H
08	A, B, D, H	C, E, F, G
09	A, B, E, F	C, D, G, H
10	A, B, E, G	C, D, F, H
11	A, B, E, H	C, D, F, G

Split	Boosted in orientation a	Boosted in orientation b
12	A, B, F, G	C, D, E, H
13	A, B, F, H	C, D, E, G
14	A, B, G, H	C, D, E, F
15	A, C, D, E	B, F, G, H
16	A, C, D, F	B, E, G, H
17	A, C, D, G	B, E, F, H
18	A, C, D, H	B, E, F, G
19	A, C, E, F	B, D, G, H
20	A, C, E, G	B, D, F, H
21	A, C, E, H	B, D, F, G
22	A, C, F, G	B, D, E, H
23	A, C, F, H	B, D, E, G
24	A, C, G, H	B, D, E, F
25	A, D, E, F	B, C, G, H
26	A, D, E, G	B, C, F, H
27	A, D, E, H	B, C, F, G
28	A, D, F, G	B, C, E, H
29	A, D, F, H	B, C, E, G
30	A, D, G, H	B, C, E, F
31	A, E, F, G	B, C, D, H
32	A, E, F, H	B, C, D, G
33	A, E, G, H	B, C, D, F
34	A, F, G, H	B, C, D, E

I PREDICTION PROTOCOL AND DIAGNOSTICS

Prediction targets. A target consists of one model, one set with at least three instructions, and one arm. Targets come from deployment and generous-reference composition, the central comparison with direction seed 0, and the transfer experiments. For each target, a predictor outputs two probabilities. The first, p_J, is the probability that an answer is valid and passes every member. The second, p_V, is the probability that the answer is valid. We fit predictors using only measurements with at most two instructions: single-instruction calibration, decomposition cells, and composition cells with $K = 2$.

Baselines and instruction-count model (F0–F1). The constant baseline predicts each target using the model's mean over $K = 2$ cells from the same arm. The always-fail baseline sets $p_J = 0$. F1 models the effect of instruction count as

$$\operatorname{logit} p_J = \alpha + \beta(K - 1),$$

with separate parameters for each model and arm. We fit it by maximum likelihood to per-prompt outcomes at $K = 1$ and $K = 2$. An analogous model predicts p_V. Because the fit uses only two values of K, it matches both pooled rates exactly.

Single-instruction curves and delivered projection (F2). Consider a normalized shared-layer rule with member directions D, coefficients w, Gram matrix G, and dose s. The delivered projection onto member i is

$$a_i = s \, \frac{(Gw)_i}{\sqrt{w^\top G w}}, \tag{2}$$

where set-order orthogonalization uses $w = R^{-1}\mathbf{1}$ for $D = QR$. For the unnormalized calibrated sum, w contains the frozen member doses and $a_i = (Gw)_i$.

We then read member i's valid compliance p_i from its single-instruction curve at dose a_i and the relevant layer. If that layer is the member's frozen layer, we use the dose-grid curve. Otherwise, we use the layer-search curve and interpolate linearly between measured doses. A member used alone at its own layer takes its frozen-test value. We combine the member probabilities with an

equicorrelated Gaussian copula,

$$p_J = \int \phi(z) \prod_{i=1}^{K} \Phi\left(\frac{\Phi^{-1}(p_i) + \sqrt{\rho}\, z}{\sqrt{1-\rho}} \right) dz, \tag{3}$$

which we evaluate with Gauss-Hermite quadrature. We fit ρ separately for each model and arm class, using $K = 2$ outcomes. Setting $\rho = 0$ gives the independent case. We estimate validity from the members' validity curves at the delivered norm.

Generic damage and partner effects (F3–F4). F3 adds the effect of an off-target component. For member i, we define its off-target norm as

$$b_i = \sqrt{\max(n^2 - a_i^2, 0)},$$

where n is the total norm and a_i comes from Equation 2. We then replace p_i with

$$p_i + \hat{G}(b_i/\text{onset}_i),$$

clipped to $[0, 1]$. Here $\hat{G}$ is a non-increasing isotonic fit to the random-component damage measured in the decomposition. F4 also adds the measured partner effect $\hat{P}_{ij}$ of each partner j. Each effect is weighted by that partner's share of the squared off-target norm. F3 and F4 apply only to models with a decomposition.

Scoring and transfer. We score each predictor family in two ways: the Brier score over test prompts, and reliability across ten bins of p_J. To test transfer across instruction families, we hold out one IFEval family at a time. We remove every decomposition cell and every $K = 2$ set that contains it, refit, and then predict the targets from that family. For the transfer experiments, we use the Qwen2.5-32B and IFBench calibrations. We pool three fitted parts across the smaller models: the copula correlation ρ, the generic-damage function $\hat{G}$, and the partner effects $\hat{P}$. Larger-model predictions cover Qwen2.5-32B only.

Holding out instruction families changes little. Appendix Q reports the pooled predictor scores. Here we test how strongly those scores depend on the fitted instruction families. We remove each of 8 families in turn, refit the predictors, and evaluate only targets from the held-out family. A target whose set contains several families is predicted once for each of them. This population differs from the one in Appendix Q, so the full-fit scores below differ from the scores reported there. Across 34,323 held-out predictions in 166 sets, the Brier scores change as follows: the instruction-count fit from 0.0131 to 0.0136, the single-instruction curve fit from 0.0146 to 0.0149, the generic-damage fit from 0.0126 to 0.0131, and the partner fit from 0.0148 to 0.0143. The always-fail baseline remains at 0.0167. On Qwen2.5-32B, the same procedure changes the instruction-count fit by 0.0023 and the generic-damage fit by 0.0010. These small shifts indicate that the scores depend little on which instruction families are fitted, within the populations we test.

The predicted probabilities are not well calibrated. Calibration is poor even though the paired comparisons show gains over matched baselines. Take the generic-damage predictor on the main population. Its lowest reliability bin contains 2,283 targets and is close to calibrated: predicted success is 0.003, and observed success is 0.001. At higher predicted probabilities, the model consistently over-predicts: 0.142 versus 0.073, 0.244 versus 0.084, 0.359 versus 0.200, and 0.442 versus 0.186. Thus, the predictors contain useful relative signal. But their probabilities should not be read as calibrated success rates. The matched comparisons support improvement over their baselines. They do not establish an ordering among predictor families.

J Composition geometry and ridge diagnostics

How we place and scale member edits. Each instruction contributes the layer and dose selected during its frozen installation calibration. Shared-layer rules choose the most common member layer, with ties going to the smaller layer index. They use the arithmetic mean of the member doses. We read every member direction at this shared layer, even when that instruction was calibrated at another

Table 19: Ridge search of the corrected rule per model. Each cell is pooled mean valid member success (percent) at one ridge value. The search uses development prompts 200 to 239, the first five full-pool sets at $K = 4$ and $K = 8$, and the deployment dose. The chosen ridge is in bold. It maximizes the pooled score, with ties going to the larger ridge. Validity uses the automatic guard alone.

Model	10^{-4}	10^{-3}	10^{-2}	10^{-1}	1	Cells
Qwen2.5-7B	**25.6**	25.5	24.8	25.3	25.5	10
Gemma-3-4B	**27.0**	26.9	26.6	26.8	26.9	10
Llama-3.1-8B	25.6	25.5	**25.8**	25.7	25.3	10

layer. Fixed-norm addition sums the directions and normalizes the result to the mean dose. The calibrated sum instead weights each unit direction by its own frozen dose and keeps the norm of the resulting sum. Layer-separated composition keeps each member's own layer and dose. When several members use the same layer, it adds their contributions. The hybrid applies this layer-separated rule to frozen reinforcement settings. It also states the instruction list in the prompt.

Geometry of normalized composition. Let D contain the original unit directions, let $G = D^\top D$ be their Gram matrix, let w be the coefficient vector, and let s be the dose. A normalized relative edit is

$$e = s\,\frac{Dw}{\sqrt{w^\top G w}}, \qquad D^\top e = s\,\frac{Gw}{\sqrt{w^\top G w}}. \tag{4}$$

The second expression gives the edit's projection onto each original direction. If the basis is orthonormal and all coefficients are equal, every basis direction receives projection $s/\sqrt{K}$. After Gram–Schmidt, however, these equal projections apply to the new orthonormal basis rather than to the original instruction directions.

The geometry-corrected rule chooses

$$w = (G + \epsilon I)^{-1} t,$$

where t is the requested projection profile. We use either a uniform profile or a priority profile that weights one designated member three times more than the others. Before final normalization, the resulting projection on the original directions is

$$G(G + \epsilon I)^{-1} t.$$

This equals the requested profile t only when there is no ridge regularization and G has full rank. Thus, regularization trades exact profile matching for numerical stability.

Selecting the ridge. The corrected rule in Section 2 introduces one additional composition parameter, the ridge ϵ. We select it once per model. The search uses development prompts 200–239, the first five full-pool sets at both $K = 4$ and $K = 8$, and the deployment dose. We test

$$\epsilon \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}.$$

We choose the value that maximizes pooled mean valid member success. Ties go to the larger ridge. The priority variant reuses the same selected value. This search uses only the automatic guard for validity. Table 19 reports the full search for each composition model.

How difficult is the geometric correction? Table 20 measures the geometry of the direction sets used by the composition experiments for each model and instruction count. We report two things. The condition number and the off-diagonal cosines show how far the Gram matrix departs from the identity. The coefficient norm shows how large a coefficient vector the corrected rule must produce. We normalize this coefficient norm by the norm of the requested profile. A value near one means that the correction changes the requested profile only mildly. A large value means that the rule must strongly amplify coefficients, usually because the Gram matrix is close to singular. Then small changes in the member directions can cause large changes in the solved coefficients. Pan et al. (2026) connect spillover to this same Gram geometry. These diagnostics therefore measure the quantities that the correction is designed to address, on our own sets and instruction counts.

Table 20: Geometry of the composition sets at the shared edit layer, per model and instruction count K, on the full-pool sets. Sets: admitted sets measured. $\kappa(G)$: condition number of the Gram matrix of member directions (median and maximum over sets). $|\cos|$: median absolute off-diagonal cosine within a set, then median over sets. $\|w\|/\|t\|$: median ratio of the solved coefficient norm to the requested profile norm. It is empty for a model with no chosen ridge.

Model	K	Sets	$\kappa(G)$, median	$\kappa(G)$, max	$\lvert\cos\rvert$, median	$\|w\|/\|t\|$, median
Qwen2.5-7B	2	30	2.5	4.9	0.425	0.70
Qwen2.5-7B	3	30	4.5	34.2	0.409	0.54
Qwen2.5-7B	4	30	5.6	39.0	0.399	0.47
Qwen2.5-7B	6	30	8.2	41.1	0.342	0.41
Qwen2.5-7B	8	30	34.2	65.7	0.385	0.39
Qwen2.5-7B	12	30	80.5	97.9	0.405	0.33
Qwen2.5-7B	15	1	92.1	92.1	0.404	0.26
Gemma-3-4B	2	30	12.3	453.1	0.849	0.54
Gemma-3-4B	3	30	50.1	805.6	0.877	0.39
Gemma-3-4B	4	30	64.2	928.8	0.845	0.37
Gemma-3-4B	6	30	182.2	7039.9	0.838	0.48
Gemma-3-4B	8	30	291.5	3376.3	0.814	0.71
Gemma-3-4B	12	30	781.1	1160.5	0.818	0.93
Gemma-3-4B	15	1	1041.0	1041.0	0.838	1.17
Llama-3.1-8B	2	30	4.0	8.1	0.599	0.62
Llama-3.1-8B	3	30	7.3	28.4	0.537	0.49
Llama-3.1-8B	4	30	11.0	46.3	0.578	0.40
Llama-3.1-8B	6	30	20.3	59.1	0.548	0.35
Llama-3.1-8B	8	30	43.8	79.0	0.584	0.34
Llama-3.1-8B	12	30	103.4	146.9	0.601	0.28
Llama-3.1-8B	15	1	126.7	126.7	0.566	0.25

K Quality evaluation and human validation

K.1 LLM Judge

Judge model and access. We use Claude Sonnet 5 (Anthropic) as the primary quality judge, with extended thinking disabled.

What the judge sees. The judge receives only the base task and the generated answer. It never sees the instruction sentences, experimental arm, model, or prompt condition. This keeps the evaluated instruction itself out of the judge input.

Judge rubric. We ask: "Does this answer address what the prompt asked for, ignoring formatting, style, case, length, language, and wording rules? Answer yes, no, or unsure, then one sentence." We count unsure answers and refusals as invalid and report refusals separately. The rubric explicitly lists language among the ignored properties because the judge never sees a requested response language.

Which answers we judge. Full validity requires both the automatic guard and the judge. An answer that fails the guard is therefore invalid, whatever the judge says. We normally send only guard-valid answers for judgment. The exception is a seeded sample of guard-invalid answers used only to measure agreement with the guard.

We judge every eligible answer in four families: the single-instruction frozen tests (Section 3), the deployment composition cells (Section 4), the two-axes experiment and both priority comparisons (Appendix M), and the transfer cells. The remaining test families use fixed prompt subsets. For each set, we take the first positions from a permutation seeded by the set identifier. Random-component experiments instead seed by the primary instruction. We use the same positions across all arms and cells for that set, so comparisons stay paired.

The central comparison judges 200 of its 1,000 prompts. The generous reference (Section 4) judges 100 prompts per set. So do the prompt controls and the revision experiments (Section 5). For these cells, judge-based results use only those judged prompts. Next to them, we report guard-only results on the same subset and on the full prompt set. Development cells and exact repeats use the guard alone. Appendix F adds judged direction-seed comparisons. We select all single-instruction settings with the guard alone. We never reselect them after applying the judge.

Table 21: Judge checks on the audited single-instruction frozen tests. The tests cover four arms: unedited, frozen edit, input-dependent coefficient, and installation settings reused with the instruction in the prompt. Requested answers are those that pass the automatic guard. Agreement with the guard counts a yes verdict on a guard-valid answer and a no or unsure verdict on a guard-invalid one. The last two rows come from the human audit of two-instruction answers (Appendix K.2).

Check	Value
Requested answers	383,002
Answers judged	383,002 (100.0%)
Answers judged twice with the same verdict	97 of 100 (97.0%)
Agreement with the guard	84.1%
Refusals	341
Human-audit sample, 3 raters	200
Human majority agreeing with the judge	80.0% ($\kappa = 0.60$)

Table 22: Judge verdicts against guard outcomes for the answers of Table 21. Guard-invalid answers are judged only in a seeded sample drawn for this check. Validity never counts them as valid.

Guard outcome	Judge yes	Judge no	Judge unsure	All
Guard valid	321,897	55,258	5,847	383,002
Guard invalid	61	417	22	500

Judge checks. Table 21 reports four checks: how many requested answers the judge scored, how often repeated judgments agree, how often judge verdicts agree with the automatic guard, and the number of refusals. Table 22 crosses judge verdicts with guard outcomes.

K.2 Human validation

Raters and blinding. 3 authors volunteered as raters and consented to the use of their ratings. The interface showed only an anonymous item identifier, the base task, and the generated answer. It hid the model, experimental arm, target instructions, and judge verdict.

Audit sample. The audit contains 200 answers in two equal strata: judge-accepted and judge-rejected. The sample is balanced across Qwen2.5-7B, Gemma-3-4B, and Llama-3.1-8B, and across the prompt-only and hybrid arms. Every answer comes from a guard-valid two-instruction deployment test and has an existing yes/no judge verdict. We exclude response-language constraints. We also require every letter in the task and the answer to have a Latin Unicode name. The audit therefore covers Latin-script answers from the prompt-only and hybrid arms of the three smaller composition models at two instructions. Its balanced judge strata also differ from the natural population proportions.

Human questions. Each rater records three observations using the following wording:

- "Does it answer what was asked?" Hint: "Ignore lowercase, missing titles, odd length, or any writing rule it was told to follow." Options: "Yes", "Partly", "No", "Not sure".
- "Any wrong facts?" Hint: "Only what you can tell by reading. You are not expected to look anything up." Options: "None spotted", "Some", "Not sure".
- "Is the writing broken?" Hint: "It repeats itself, stops in the middle of a sentence, or turns into nonsense." Options: "No", "Yes", "Not sure".

Human decision rules. Each item has one final rating from each rater. We map the observations to three acceptance rules. The permissive relevance rule accepts both "Yes" and "Partly" for task response. The strict relevance rule accepts only "Yes". The combined quality rule requires "Yes" for task response, "None spotted" for factual errors, and "No" for broken writing. Under the combined rule, any "Not sure" response gives an unsure label, whatever the other responses are. A factual rating of "None spotted" means only that the rater noticed no error. It does not verify factual correctness.

For each rule, we first assign each rater a yes/no/unsure label and then take the majority label across raters. Ties become unsure. When comparing with the binary judge, we map every outcome other than yes to nonacceptance.

Table 23: **Human agreement depends on the acceptance rule.** The same 200 answers and 3 raters' final ratings are used throughout. The sample is equally divided between judge-accepted and judge-rejected answers. Agreement and Cohen's κ compare the binary judge verdict with the human majority derived under each rule. The last two columns condition on the judge's verdict. Human unsure outcomes count as nonacceptance. Percentages describe this balanced audit sample.

Human rule	Agreement (%)	κ	Accept given judge yes (%)	Do not accept given judge no (%)
Relevance: Yes or Partly	51.0	0.02	100	2
Relevance: Yes only	61.0	0.22	96	26
Combined quality	80.0	0.60	78	82

Agreement supports a broader quality interpretation. Table 23 reports agreement under each rule. Under the combined quality rule, the human majority agrees with the judge on 80.0% of answers, with Cohen's $\kappa = 0.60$. Humans accept 78% of judge-accepted answers. They do not accept 82% of judge-rejected answers, counting unsure outcomes as not accepted. Agreement falls to 61.0% ($\kappa = 0.22$) under strict relevance and to 51.0% ($\kappa = 0.02$) under permissive relevance. Under permissive relevance, humans accept 100% of judge-accepted answers. They also accept almost all judge-rejected answers: only 2% are not accepted. These differences suggest that the judge tracks broad answer quality better than pure task relevance. They do not validate it as a relevance-only measure.

Inter-rater agreement under the combined quality rule is Fleiss' $\kappa = 0.42$, with pairwise Cohen's κ ranging from 0.41 to 0.43. These inter-rater statistics keep all three labels. The judge-human comparison is binary. The inter-rater statistics measure disagreement among raters and do not provide an upper bound on agreement with the majority label. Zheng et al. (2023) also report high judge-human agreement. Their task and agreement definition differ from ours, so the rates are not directly comparable.

K.3 Additional judge diagnostics

A second judge gives a consistency check. A second model, OpenAI GPT-5.6-terra, uses the same instruction-blind rubric. The two judges agree on binary acceptance for 89.0% of evaluated answers on Gemma-3-4B (11,731 arm-position observations), 84.7% on Qwen2.5-7B (11,460), and 88.3% on Llama-3.1-8B (11,603). Cohen's κ for this binary agreement is 0.64, 0.60, and 0.58, computed from the agreement and the two acceptance rates. This is close to the κ of 0.60 between the primary judge and the human majority under the combined quality rule (Appendix K.2). This comparison treats yes as acceptance and all other verdicts as nonacceptance. It does not require identical three-way labels. The primary and second judges accept 82.2% and 80.3% of answers on Gemma-3-4B, 73.9% and 75.1% on Qwen2.5-7B, and 85.3% and 81.7% on Llama-3.1-8B.

Full-position rescoring covers every test position. Table 2 recomputes all four scoring rules on all 200 test positions of each of the 600 frozen-test cells. This covers 240,000 arm-position observations. All 231,816 guard-valid observations already have a first-pass judge verdict, with no missing or conflicting required labels. A guard-invalid answer has full validity zero and therefore needs no judge verdict.

L Matched decomposition of composition loss

The decomposition studies capital-word frequency on Llama-3.1-8B at layer 27 over 200 test prompts. A primary qualifies only if instance 0 has both a usable guard-only installation and a defined development need. Capital-word frequency is the only instruction on the smaller composition models that satisfies both conditions. The all-lowercase instruction's instance 0 passes the frozen-test usability rule but has no usable dose on its development grid. For each model, the selection considers up to six eligible instructions. It ranks them by onset-to-need ratio, from largest to smallest, with ties broken by name. The first three become core primaries.

Matched component design. Appendix A defines need and onset. When onset is undefined, we substitute 1.6 and define

$$\text{mid} = \sqrt{\text{need}\,\text{onset}}.$$

For primary projection a, let

$$b_{\max}(a) = \sqrt{\max((1.2\,\text{onset})^2 - a^2, 0)}.$$

We evaluate three core points:

$$(\text{need}, 0.6b_{\max}(\text{need})), \qquad (\text{mid}, 0.6b_{\max}(\text{mid})), \qquad (\text{mid}, b_{\max}(\text{mid})).$$

Every component class uses the same primary projection and added-component norm at each core point.

For the eligible primary, compatible partners cause more loss than matched isotropic components. At each core point, we hold the primary edit fixed and add a second component with the same magnitude as a real partner component. We decompose the resulting drop in the primary's valid compliance into two terms. The *generic* term measures the effect of an isotropic component of the same size, averaged over 100 random draws. The *partner-specific* term is the remaining effect after subtracting this generic component. Both terms use the automatic guard alone.

Every cell generates 200 answers. The generic-damage and partner-specific comparisons use the same 100 fixed positions. Generic damage is -4.5 [-9.0, 0.1], -2.3 [-6.7, 2.5], and -4.1 [-8.9, 1.0] points at the three core points. Every interval contains zero. The partner-specific effects are larger: -8.9 [-12.6, -5.4], -9.3 [-13.2, -5.7], and -21.1 [-26.2, -16.1] points. No interval contains zero. In this primary, compatible instruction directions therefore cause more compliance loss than isotropic components of the same size.

The largest added component causes the clearest interference. The partner component has relative magnitude 0.179, 0.170, and 0.283 at the three core points. The largest component produces the largest partner-specific loss. The two smaller components are similar in size, and their interference intervals overlap. The data therefore separate only the largest component from the two smaller ones.

Other real directions are also harmful. Relative to matched isotropic components, other real instruction directions have residual effects of -13.4, -12.5, and -24.7 points. Their point estimates show even larger losses than the compatible partners at the same component sizes (Appendix G). This result suggests that the excess damage is not unique to the selected compatible partners.

L.1 Extending the decomposition to already-satisfied instructions

The eligibility rule above leaves only one primary. To test a broader population, we drop the usable-installation requirement but still require a defined development need. This yields 3 core primaries on each of Qwen2.5-7B, Gemma-3-4B, and Llama-3.1-8B. Every one of these primaries is already satisfied in the guard-only installation bank. This extension therefore asks how added components disturb instructions that the model already follows. It does not represent a broader population of failed installations.

We keep the same three core-point definitions and component classes. Validity still uses the automatic guard. Each core point uses 20 isotropic draws rather than 100. The results below pool 9 core cells per model over 100 fixed prompt positions. Realized edit magnitudes differ across primaries. All three Qwen primaries use the fallback value of 1.6 for a missing onset.

Generic damage becomes substantial in the broader population. Pooled generic damage is -42.1 [-46.3, -38.1] points on Qwen2.5-7B, -93.9 [-96.1, -91.6] on Gemma-3-4B, and -33.6 [-37.0, -30.3] on Llama-3.1-8B. None of the intervals contains zero. Matched isotropic components therefore cause substantial compliance loss at these operating points. The small generic effect for the capital-word primary does not extend to this broader population. Instruction identity, layer, and delivered edit magnitude also change across primaries. This comparison therefore does not identify which factor causes the larger damage.

Partner-specific effects differ across models. Pooled partner-specific interference is -11.2 [-14.6, -7.4] points on Qwen2.5-7B and -8.5 [-11.3, -5.5] on Llama-3.1-8B. Both intervals

Table 24: Selected quality-aware steering studies. Outcomes differ in both their criteria and their aggregation. Same-answer quality evaluation is already used in prior work.

Study	Intervention	Settings	Quality assessment	All criteria in one answer
Stolfo et al. (2025)	Instruction vectors from with/without activation differences	Calibrated on held-out examples	Instruction-blind judge	No: quality scored separately from compliance
Subbiah et al. (2026)	Writing-preference trait vectors, several composition rules	Per-setting tuning reported as necessary	Coherence beside trait expression	No: a tradeoff between two separate measures
Kang et al. (2026)	Additive composition of attribute vectors	Layer and strength chosen per attribute from single-attribute runs	Output relevance as one metric	No: metrics averaged over generations, then combined by harmonic mean
Wu et al. (2025)	Concept-steering interventions	Strength selected on half of the instructions, tested on the other half	Concept, relevance, and fluency scores	Yes: harmonic mean per answer, then averaged
This paper	Instruction directions under several composition rules	Frozen from single-instruction calibration	Automatic guard and instruction-blind judge	Yes: every requested instruction and the quality checks in the same answer

lie below zero. Gemma-3-4B instead gives a positive residual of 14.9 [13.9, 15.8]. Its compatible partner components are less harmful on average than matched isotropic components.

On Gemma, the isotropic components already reduce valid compliance to at most 2.3%. This leaves little room for an additional negative partner effect. This floor limits how negative the residual can become but does not remove the observed positive difference. Other real directions match the sign of the pooled partner term on every model. Their effects are -9.3 points on Qwen2.5-7B, 15.1 on Gemma-3-4B, and -11.0 on Llama-3.1-8B. These averages cover 3 instructions per model and vary across core points. They therefore do not support a uniformly negative partner effect or a universal partner-specific explanation for composition loss.

Direction geometry varies across models. For two-instruction sets on Gemma-3-4B, the median absolute off-diagonal cosine of the Gram matrix is 0.849 across 30 sets. The median condition number is 12.3, with a maximum of 453.1. Qwen2.5-7B has a lower median absolute cosine of 0.425, and Llama-3.1-8B has 0.599. For Gemma-3-4B, the corrected composition rule produces coefficient vectors with norms smaller than the requested profile norm. The median ratio of the two norms is 0.541. These measurements connect the decomposition results to the direction geometry analyzed further in Appendix J.

M Practical baselines and controls

Prompt wording changes little. For each model, we select the structured template with higher pooled development success. We keep the "Follow all of these requirements" template unless the other template improves by at least one point. This exception applies to all three models, so each model selects the "Follow all of these requirements" template. We pool all instruction counts with an available set: two, three, four, six, eight, and twelve. The selected template therefore need not be the best one at two instructions alone. This happens on Gemma-3-4B, where the two-instruction rates are 59.5% for "obey every rule" and 56.5% for "Follow all". It also happens on Llama-3.1-8B, with 55.0% and 54.2%. Table 28 gives the test-prompt contrasts.

Simple deterministic transforms can enforce a narrow class of constraints. Deterministic post-processing changes no model weights and requires no search. It is the cheapest alternative we test. It supports 7 constraints: lower case, upper case, no comma, a required end phrase, quotation marks,

Table 25: Composition on sets selected because each member has a usable reinforcement setting under the automatic guard. Values are joint valid success (percent, with a 95% interval over sets and prompts) on test prompts. The frozen edits take every layer and dose from single-instruction calibration. The tuned edits are the generous reference, which searches the dose scale of each set on development prompts and then tests the chosen setting once. The tuned cells judge 100 of the 200 prompts of each set. The other cells judge every guard-valid answer.

Setting	$K = 1$	$K = 2$	$K = 3$	$K = 4$	Validity, $K = 4$
Qwen2.5-7B					
Sets	5	0	0	0	
Prompt only	50.9 [36.8, 61.9]	–	–	–	–
Prompt plus the frozen edits	66.0 [59.9, 71.7]	–	–	–	–
Prompt plus the per-set tuned edits	–	–	–	–	–
Gemma-3-4B					
Sets	9	10	10	5	
Prompt only	47.8 [33.8, 59.2]	19.8 [7.5, 33.7]	5.9 [1.6, 11.7]	1.3 [0.2, 2.9]	78.0
Prompt plus the frozen edits	69.7 [62.9, 76.2]	41.1 [26.5, 53.9]	20.5 [9.8, 31.6]	6.2 [0.7, 14.9]	18.9
Prompt plus the per-set tuned edits	–	41.9 [28.6, 54.5]	31.3 [21.2, 41.4]	18.4 [9.4, 27.6]	64.2
Llama-3.1-8B					
Sets	11	0	0	0	
Prompt only	60.0 [54.2, 65.5]	–	–	–	–
Prompt plus the frozen edits	72.1 [66.6, 77.1]	–	–	–	–
Prompt plus the per-set tuned edits	–	–	–	–	–

a title line, and a postscript line. We apply it to the outputs of the instruction-list prompt. Each transformed answer keeps the guard and judge verdicts of its original answer. Across all 30 two-instruction sets, it changes joint valid success by 3.6 [0.9, 7.5] points on Qwen2.5-7B, 8.4 [2.2, 16.3] on Gemma-3-4B, and 6.7 [2.0, 13.2] on Llama-3.1-8B. On the 17 sets that contain at least one supported constraint, the changes are 6.4 [1.8, 12.8], 14.8 [4.4, 27.6], and 11.8 [4.0, 22.8] points, in the same model order. All six intervals exclude zero.

The transforms are not perfectly reliable. Of 4,400 checks per model, 215 fail on Qwen2.5-7B, 7 on Gemma-3-4B, and 4 on Llama-3.1-8B. All failures are on the two case constraints. This baseline is useful only where a direct text transform can enforce the constraint. It does not cover the rest of the instruction pool.

Learned steering tokens provide a separate control mechanism. We reimplement steering tokens (Radevski et al., 2026) and train 6 behavior tokens plus one joining token for each composition model. Our version uses a learning rate of 10^{-2} rather than 10^{-4}, selected on held-out training prompts. Training targets are each model's own greedy outputs rather than answers generated by Qwen3-30B-A3B-Instruct. Each behavior uses 10,000 examples instead of 50,000, one to three available instances rather than ten paraphrases, and a 1,536-token cap rather than a 256-token generation cap. We filter single-turn Smoltalk prompts for existing constraints with pattern matching instead of the published model-based splitting procedure. Table 27 reports the results.

Stating an instruction and injecting it are not interchangeable. The two-axis experiment varies how many instructions appear in the prompt and how many are supplied through edits. We use one fifteen-instruction set and score its four fixed targets (Appendix H). When the prompt states none of the target instructions, joint valid success is zero on all three models, at every injected count. With neither prompt nor edits, it is 0.0% on Qwen2.5-7B, 0.0% on Gemma-3-4B, and 0.0% on Llama-3.1-8B. Injecting all fifteen instructions also gives 0.0% in every arm on all three models.

Among prompts without edits, stating only the target instructions gives the highest target success. On Gemma-3-4B, removing those four prompted instructions changes success by -23.0 [-29.0, -17.5] points. Expanding the prompt to eight instructions changes it by -11.0 [-17.5, -4.5] points. With the four targets in the prompt, removing the four target edits changes success by -8.5 [-14.5, -2.5] points on Qwen2.5-7B and -11.5 [-19.0, -3.5] on Llama-3.1-8B. Reinforcement edits therefore help this prompt on those two models. This experiment isolates the two delivery axes within one fifteen-instruction set. It does not estimate a general scaling law for larger sets.

Prompted priorities produce stronger effects than priority edits. The priority experiment asks the model to favor one subset of instructions over another across 35 partitions of a fifteen-instruction set (Appendix H). We measure how strongly each arm follows the requested priority relative to a null edit that specifies no priority. Stating the priority in the prompt produces a clear effect on all

Table 26: **Checker-guided revision.** Composition sets of two instructions on the predefined test prompts, pooled over sets with equal weight. Each block gives joint success (%) after each pass and the paired change from first to last pass (points), with 95% bootstrap intervals over sets and prompts. *Guard and judge* applies full validity on the judged positions. *Guard alone* applies only the guard on the same positions. *Checkers alone* uses member checkers on all test prompts, with no validity rule. Revision raises joint valid success on all three models, at the cost of extra generations.

Readout	Qwen2.5-7B	Gemma-3-4B	Llama-3.1-8B
Sets	30	30	30
Guard and judge, first pass	46.1 [36.2, 55.6]	50.0 [36.9, 62.6]	46.0 [34.1, 57.3]
Guard and judge, second pass	51.9 [41.4, 62.0]	55.7 [42.9, 67.8]	54.8 [43.0, 65.9]
Guard and judge, third pass	52.7 [42.3, 62.7]	56.9 [44.2, 69.0]	56.7 [44.9, 67.9]
Guard and judge, third pass minus first	6.7 [3.7, 10.3]	6.9 [2.9, 12.2]	10.7 [6.9, 14.9]
Guard alone, first pass	59.0 [47.0, 70.3]	56.5 [41.7, 70.6]	51.4 [38.5, 63.7]
Guard alone, second pass	66.7 [54.2, 78.3]	63.4 [49.1, 76.9]	62.5 [49.2, 74.8]
Guard alone, third pass	67.8 [55.3, 79.3]	64.9 [50.7, 78.2]	64.7 [51.4, 77.0]
Guard alone, third pass minus first	8.8 [5.2, 13.1]	8.4 [3.8, 14.2]	13.3 [8.7, 18.3]
Checkers alone, first pass	60.6 [48.4, 72.1]	56.7 [42.1, 70.8]	51.3 [38.3, 63.5]
Checkers alone, second pass	68.3 [55.6, 80.0]	63.6 [49.2, 76.9]	63.2 [50.0, 75.3]
Checkers alone, third pass	69.4 [56.8, 81.1]	65.0 [50.7, 78.1]	65.8 [52.5, 77.8]
Checkers alone, third pass minus first	8.9 [5.3, 13.1]	8.3 [4.0, 13.7]	14.5 [9.9, 19.5]
Mean generations per answer	1.71 [1.48, 1.96]	1.80 [1.53, 2.08]	1.86 [1.62, 2.11]

Table 27: Our reimplementation of steering tokens on the 94 official IFEval prompts whose instructions all belong to the trained behaviors. Each cell gives prompt-level strict accuracy / loose accuracy / degeneration rate, in percent. These prompts state their instructions in the task, so the judge cannot be blind to them. Degeneration is the only quality measure. Rates pool prompts with one, two, or three instructions.

Arm	Qwen2.5-7B	Gemma-3-4B	Llama-3.1-8B
Prompt as given	92.6 / 94.7 / 16.0	88.3 / 91.5 / 8.5	86.2 / 87.2 / 6.4
Prompt and the behavior tokens	81.9 / 86.2 / 17.0	88.3 / 94.7 / 7.4	90.4 / 91.5 / 6.4
Prompt and IFEval's sentence again	88.3 / 90.4 / 14.9	90.4 / 93.6 / 8.5	84.0 / 85.1 / 7.4
Prompt, the sentence, and the tokens	84.0 / 86.2 / 18.1	91.5 / 95.7 / 7.4	89.4 / 90.4 / 5.3

three models: 13.2 [11.2, 15.2] points on Qwen2.5-7B, 1.8 [0.9, 2.7] on Gemma-3-4B, and 4.8 [3.5, 6.1] on Llama-3.1-8B. None of these intervals contains zero.

Encoding the same priority through activation edits produces smaller effects. The edit tracks the requested priority on Gemma-3-4B by 1.5 [0.9, 2.1] points and on Llama-3.1-8B by 1.4 [0.7, 2.1]. On Qwen2.5-7B the effect is -0.5 [-1.2, 0.1], with an interval containing zero.

We repeat the comparison on one split drawn from instructions outside the original priority pool (Appendix H). On this new pool, prompting tracks the requested priority only on Qwen2.5-7B, with an effect of 3.5 [0.5, 6.5] points. The corresponding effects are -0.3 [-3.3, 2.7] on Gemma-3-4B and -1.1 [-4.1, 1.8] on Llama-3.1-8B. The prompted-priority advantage therefore does not carry over to the new pool.

M.1 Revision and matched retry

Generic retries recover much of the revision gain. Table 29 reports the checker-guided revision experiment, and Table 26 gives its full guard-only and checker-only readouts. Table 5 adds a matched generic-retry control and evaluates both arms under the same procedure. Its diagnostic arm is a separate run from Table 29, so the two tables differ slightly. For diagnostic revision versus generic retry, first-pass joint valid success is 46.0% versus 46.2% on Qwen2.5-7B, 50.0% versus 50.3% on Gemma-3-4B, and 46.2% versus 45.7% on Llama-3.1-8B. The final rates are 52.6% versus 50.1% on Qwen2.5-7B, 56.8% versus 54.5% on Gemma-3-4B, and 56.9% versus 51.2% on Llama-3.1-8B.

Table 28: Joint valid success of each prompt wording on the two-instruction composition sets, on the predefined test prompts. Sets are pooled with equal weight. This is a separate run from the central comparison in Section 4, so the instruction-list values differ slightly. The top block gives percentages. The bottom block gives the paired difference from the instruction list in points. Each cell has a 95% bootstrap interval over sets and prompts. Validity uses the automatic guard alone, on all test prompts. Only emphasis on Qwen2.5-7B has an interval that excludes zero. Section 5 reports the judged contrasts on the judged positions, where every interval contains zero.

Prompt	Qwen2.5-7B	Gemma-3-4B	Llama-3.1-8B
Sets	30	30	30
Instruction list	59.2 [47.3, 70.5]	56.5 [41.9, 70.5]	51.0 [38.1, 63.3]
Instruction list, every instruction emphasized	64.1 [51.6, 75.5]	59.5 [45.2, 72.9]	53.3 [39.9, 66.0]
Instruction list in reverse order	55.0 [42.2, 66.9]	57.2 [42.0, 71.5]	50.9 [37.8, 63.2]
Numbered list, the model's chosen template	62.3 [50.8, 73.3]	57.3 [42.8, 71.2]	54.4 [41.6, 66.4]
Emphasis minus the instruction list	5.0 [1.4, 9.0]	3.0 [−1.4, 7.6]	2.3 [−1.0, 5.9]
Reverse order minus the instruction list	−4.1 [−12.1, 2.0]	0.7 [−4.6, 6.7]	−0.1 [−2.8, 2.8]
Chosen template minus the instruction list	3.2 [−1.4, 8.9]	0.8 [−1.5, 3.6]	3.4 [−0.8, 9.2]

Table 29: **Checker-guided revision on instruction pairs.** Joint valid success after the first and final pass, with the paired change in percentage points and its 95% bootstrap interval. Each model uses 30 sets and the same 100 judged positions per set across passes. The final column reports mean generations per answer. Table 26 gives the full guard-only and checker-only readouts.

Model	First pass	Final pass	Change [95% interval]	Generations
Qwen2.5-7B	46.1%	52.7%	6.7 [3.7, 10.3]	1.71
Gemma-3-4B	50.0%	56.9%	6.9 [2.9, 12.2]	1.80
Llama-3.1-8B	46.0%	56.7%	10.7 [6.9, 14.9]	1.86

The matched comparison does not share identical first-pass outputs. Although both arms use the same greedy first-pass procedure, their generated text differs on 2,957 prompts on Qwen2.5-7B, 3,131 on Gemma-3-4B, and 2,544 on Llama-3.1-8B, counting only prompts whose first answer is final in both arms. We therefore pair the analysis by set and prompt rather than by identical initial text. Both arms receive a failure signal and may generate additional answers. But their final contrast still depends on these different first-pass outputs. The equal pass cap also does not imply equal realized compute, because stopping times differ.

N Checker success does not guarantee answer quality

This appendix examines answers that pass every deterministic checker but still vary in overall quality. We use the human audit from Appendix K.2. We keep only answers that pass every instruction checker in their set. Of the 200 audited answers, 124 meet this condition. These examples show why checker compliance, judge acceptance, and human acceptance measure different properties.

Checker success does not imply human acceptance. Table 30 compares the quality judge with the human majority under the combined quality rule, using only answers that pass every checker. Human raters accept 65 of these answers, reject 46, and mark 13 as unsure. Thus, even perfect checker compliance within a set does not guarantee that humans consider the answer acceptable. The table also shows where the judge and the human raters disagree. Some answers pass every checker and get a yes from the judge, but still fail the human criterion. The audit balances judge-accepted and judge-rejected examples by design. These counts therefore describe the audit sample, not population rates.

Literal checker success can damage the answer. The following two examples pass every checker in their set, pass the automatic guard, and receive a yes verdict from the quality judge. The joint metric therefore accepts both. The human majority rejects the first and is unsure about the second. Both sets contain `length_constraints:nth_paragraph_first_word`, which requires a specified paragraph to begin with a specified word. In both cases, the required word appears at the correct position but disrupts the answer.

The first example comes from Gemma-3-4B with the prompt plus layer-separated reinforcement edits (the hybrid). The instructions require paragraph four to begin with *pattern* and prohibit commas.

Table 30: Judge verdict against the rater majority under the combined quality rule, on the 124 audited answers that satisfy every instruction checker of their set. The audit has equal judge-accepted and judge-rejected strata by design. These are therefore counts within the audit, not population rates.

	Rater majority		
Judge	Accept	Do not accept	Unsure
Accept	52	9	5
Do not accept	13	37	8

```
Gerald the giraffe was sad.  His friends Leo and Sam were
mean to him.  ''You are so tall'' Leo said.  ...  Gerald
wanted to cry.  ''It's not fair'' he whispered.
Pattern.  Gerald felt a little better.  ''I will go away''
he said.  ''If you stop being mean'' Leo offered.
```

The checker accepts *Pattern* because it appears at the required location. The answer also satisfies the comma constraint. Yet the word forms a standalone sentence with no meaningful role in the story. The output therefore satisfies the literal constraints while damaging the response itself.

The second example comes from Qwen2.5-7B with the instruction list and no activation edit. This shows that the failure is not specific to steering. The instructions require the first paragraph to begin with *ear* and the response to contain a bracketed placeholder.

```
Ear the brisket out of the refrigerator and let it sit at
room temperature for about an hour before smoking.  ...
Preheat your smoker to around 225°F and prepare your wood
chips.
```

The required word appears in the correct position, so the checker records success. But it sits where the sentence needs a verb. The human majority marks this answer as unsure. Our binary human comparison maps unsure to nonacceptance.

Constraints can also compose without harming the answer. The final example comes from Gemma-3-4B with the prompt plus layer-separated reinforcement edits (the hybrid). Its set requires marked sections and limits the number of occurrences of the letter *y*.

```
Section 1
Focus is key.  People should work to improve their ability
to concentrate on tasks.  Resilience aids in overcoming
obstacles.  Adaptability to change is also important.
Honesty builds trust.
Section 2
Hard work is beneficial.  A positive attitude helps.  People
should be reliable.  Resourcefulness aids in solving issues.
Knowledge is valuable.
```

Here the constraints shape the response without displacing its content. Every checker passes, and both the quality judge and the human majority accept the answer.

Together, the three examples show why we keep constraint satisfaction separate from answer quality. Every answer passes all requested checkers, yet their usefulness differs substantially. The quality judge catches some of these failures in the broader audit. But the first two examples show that judge-based validity does not remove them completely.

O Official IFBench evaluation

We evaluate the steering settings on IFBench's official test rows, whose prompts differ from the prompts used in Section 3. We use the 145 official rows whose constraints all belong to a set of 24 IFBench constraint types. This set contains the 20 types we calibrate (Section 2). We also report the 121 rows where every constraint has a calibrated bank setting. In these rows, no member stays unedited.

Table 31: The official IFBench rows, with and without the frozen edits of their constraints. These are the benchmark's own prompts, which already state their instructions. Strict and loose are official prompt-level scores, requiring every constraint of a row. Guard is the automatic validity rate. The left block uses all 145 rows. The right block uses the 121 rows whose constraints all have a bank setting, so no member goes unedited.

	All 145 rows			121 fully calibrated rows		
Model and arm	Strict	Loose	Guard	Strict	Loose	Guard
Qwen2.5-7B, prompt	29.7%	30.3%	95.2%	23.1%	24.0%	95.0%
with edits	29.0%	29.7%	91.7%	23.1%	24.0%	90.9%
Gemma-3-4B, prompt	33.1%	38.6%	93.8%	25.6%	31.4%	92.6%
with edits	35.9%	37.2%	93.1%	28.9%	29.8%	91.7%
Llama-3.1-8B, prompt	40.7%	42.1%	89.0%	33.9%	35.5%	88.4%
with edits	40.0%	43.4%	92.4%	33.9%	38.0%	91.7%

For each row, we compare two arms. The first is the official IFBench prompt with no edit. The second is the same prompt with one reinforcement edit per calibrated constraint, each at its own frozen layer. If a constraint type has no calibrated setting, we leave it unedited. We score each row with IFBench's official prompt-level criterion, which requires all constraints in the row to pass together. Validity uses the automatic guard alone, because the official prompt itself contains the evaluated constraints.

The edits do not produce a resolved strict-accuracy change. On all 145 rows (Table 31), adding the edits changes strict accuracy by -0.7 points $[-6.2, 4.8]$ on Qwen2.5-7B, 2.8 $[-2.1, 7.6]$ on Gemma-3-4B, and -0.7 $[-6.2, 4.8]$ on Llama-3.1-8B. Every interval contains zero.

On the 121 fully calibrated rows, the corresponding differences are 0.0 $[-6.6, 6.6]$ on Qwen2.5-7B, 3.3 $[-2.5, 9.1]$ on Gemma-3-4B, and 0.0 $[-6.6, 6.6]$ on Llama-3.1-8B. The Qwen and Llama point estimates are exactly zero. Their intervals still allow both gains and losses. On this subset, Gemma-3-4B has a positive point estimate above three points. Its interval still contains zero. These results therefore do not establish either an advantage from the edits or equivalence with prompting.

This evaluation also tests transfer across prompts. The edits were calibrated on prompts other than the official IFBench rows. Any gain or loss therefore reflects both the intervention and this prompt shift.

Guard validity and loose accuracy suggest additional changes. Strict accuracy does not capture every effect of the edits. With the edits, guard validity changes from 95.2% to 91.7% on Qwen2.5-7B, from 93.8% to 93.1% on Gemma-3-4B, and from 89.0% to 92.4% on Llama-3.1-8B. Loose accuracy decreases on Qwen2.5-7B and Gemma-3-4B and increases on Llama-3.1-8B. Together, these point estimates suggest that an edit can affect guard validity or partial compliance even when strict prompt-level accuracy changes little.

P RUBRIC SENSITIVITY

Our primary judge asks whether an answer addresses the task. But the human audit in Appendix K.2 agrees with the judge most strongly under a broader rule. Under this rule, the answer must fully address the task, contain no noticed factual error, and have intact writing. We therefore test whether this broader rubric changes the single-instruction classifications in Section 3. We rejudge every frozen single-instruction answer of the four smaller models that has a verdict from the primary judge. The new judgment uses the same three questions shown to the human raters. We then derive acceptance from the three responses with exactly the same rule as the human audit. We do not ask the model for an overall judgment.

The broader rubric penalizes edited answers more strongly. Among answers scored under both rubrics (Table 32), the relevance rubric accepts 86.8% of unedited answers and 84.3% of edited answers. The audit rubric accepts 64.7% and 58.2%, respectively. Acceptance therefore falls by 22.1 points for the unedited arm and by 26.2 points for the edited arm. The broader quality criterion thus widens the single-instruction gap instead of closing it. In these frozen tests, adding factual-error and writing-quality checks removes more edited answers than unedited ones.

Table 32: The primary relevance rubric against the audit's three questions, on the 383,002 frozen-test answers scored under both. The acceptance rows cover the 116,042 unedited and 115,774 edited answers among them. Acceptance is per answer. Differences use unrounded rates. The classification rows apply the usable target and quality bounds to the same settings, without reselection. The stricter rubric costs the edited arm more acceptance than the unedited one. As a threshold gate, it admits nothing at all.

	Relevance rubric (primary)	Audit rubric (three audit questions)
Accepts, unedited arm	86.8%	64.7%
Accepts, edited arm	84.3%	58.2%
Acceptance lost, unedited	—	22.1 points
Acceptance lost, edited	—	26.2 points
Usable installations	0	0
Usable reinforcements	18	0
Already satisfied, reinforcement	96	0

The broader rubric admits no setting at all. Under the audit rubric, none of the 300 tested instances reaches the usability target in either design. The unedited prompted model also retains no already-satisfied instances. Usable reinforcements and already-satisfied instances therefore fall to zero, from 18 and 96 under the primary rubric. Usable installations are zero under both rubrics. A gate that rejects every edited setting and every unedited baseline cannot tell feasible instructions from infeasible ones at the operating threshold of Section 3. We therefore keep the primary rubric for Tables 1 and 2 and use the audit rubric only as a stricter sensitivity analysis.

The check strengthens the coverage result. Under a stricter quality criterion, the limited single-instruction coverage of Section 3 persists. It also becomes more severe for edited answers.

Factual-error judgments account for 65.4% of the additional rejections. The judge does not check facts against outside sources. These labels therefore reflect errors it notices in the answer itself, not established factual correctness. We therefore interpret the audit rubric as a stricter quality screen, not as a more accurate ground-truth evaluator.

Q PREDICTING UNSEEN COMBINATIONS

This appendix asks whether behavior on one and two instructions predicts performance on larger unseen combinations. We fit predictors using only measurements with at most two instructions. Each predictor estimates the probability that an answer is valid and satisfies every member of a target set. Each target specifies one model, set, and arm. Appendix I gives the predictor definitions.

Predictor gains use matched targets. The main population contains 7,548 targets across 151 sets. Lower Brier scores are better. The generic-damage fit scores 0.0160 [0.0130, 0.0192], the instruction-count fit 0.0171 [0.0135, 0.0209], the single-instruction curve fit 0.0174 [0.0143, 0.0206], and the partner fit 0.0180 [0.0150, 0.0211]. These marginal scores may cover different targets. The damage and partner fits are unavailable when their required decomposition was not run. We therefore do not rank them as if they shared the same population. Instead, each comparison below uses only targets shared by a predictor and its baseline. All four fitted families improve over always predicting failure, with intervals excluding zero: -0.0051 [-0.0077, -0.0027] for instruction count, -0.0047 [-0.0089, -0.0010] for the curve fit, -0.0072 [-0.0119, -0.0031] for generic damage, and -0.0052 [-0.0101, -0.0008] for the partner fit.

Rare joint success favors the always-fail baseline. Always predicting failure scores 0.0221 [0.0165, 0.0283]. Predicting each model's mean over its two-instruction cells scores 0.0483 [0.0480, 0.0487], more than twice as high. Because joint success is rare at these set sizes, predicting failure is usually correct. A low score alone therefore does not imply strong predictive value. For the same reason, the absolute gains above remain small.

The instruction-count fit transfers to a larger model; no fit transfers to IFBench. On the out-of-domain IFBench constraints (360 targets in 24 sets), no fitted family improves over always predicting failure, which scores 0.0025 [0.0003, 0.0065]. Larger-model predictions cover Qwen2.5-32B only. Across 200 targets in 40 sets, the instruction-count fit scores 0.0584 [0.0383, 0.0804], compared with

0.0882 [0.0521, 0.1281] for always fail. Their paired difference is -0.0298 [-0.0488, -0.0130], with the interval excluding zero. For transfer, we use the single-instruction calibration of each new model or benchmark. The predictor parameters learned on the smaller IFEval models stay fixed (Appendix I). These results remain conditional on the witness screen used to admit sets. One model that writes witness answers is also a transfer model. Its sets were therefore screened on tasks it could already satisfy (Appendix D).