WORD SUBSTITUTION BYPASSES SAFETY MONITORS DESPITE EXPLICIT MAPPINGS

Anonymous authors
Paper under double-blind review

ABSTRACT

Word substitution can bypass language-model safeguards by replacing sensitive terms with harmless-looking words, even in some of the recent advanced models. External safety monitors add another layer of defense, yet it remains unclear whether they can recognize harmful content even when told what the substituted words mean. To further investigate this issue, we conducted a comprehensive and systematic study of this phenomenon. Our empirical study first reveals that, across seven generation services, substitution-based attacks persist against some frontier models; seven monitors show 13.3–76.7 percentage-point alarm reductions on paired held-out inputs despite explicit meanings; and restoring questions increases alarms at least as much as restoring answers. Also, in an open-weight monitor, swapping codeword meanings while fixing vocabulary, questions, and answers yields *distinguishable hidden states but unchanged safety decisions*: mappings are linearly separable on unseen questions, while scores shift but remain below threshold. Explicit harmful concepts instead trigger some alarms. On generated answers, templates, output prefixes, and general instructions receive 92% of direct decision-position attention, with similar aggregate allocations across mappings. Demonstrations improve detection on unseen questions, and lightweight fine-tuning improves risk ranking under matched inference settings. Identical warnings yield different harmful-delivery outcomes depending on whether triggering text is withheld. Our findings show that explicit word meanings and distinguishable internal states are insufficient for reliable safety monitoring, informing how external monitors should be evaluated and improved.

1 INTRODUCTION

As large language models become more capable and widely deployed, preventing harmful assistance is a key safety requirement. Although safety alignment aims to prevent such assistance, changes in expression can bypass these safeguards (Wei et al., 2023; Yuan et al., 2024). In particular, *word substitution* replaces sensitive terms with innocuous words or codewords to elicit responses a model would otherwise refuse (Handa et al., 2025; Zhang et al., 2025; Yan et al., 2025; Yona et al., 2026). Context supplies the substituted terms' meanings, preserving harmful intent while changing its expression.

Word substitution remains effective against some frontier LLMs in our evaluation of seven recent service configurations (Appendix B). Stronger alignment within the generator can impose a safety–capability trade-off, reducing reasoning ability or other task performance (Huang et al., 2025; Chen et al., 2026). External monitors offer a complementary defense: they inspect requests or generated content without changing the generator's parameters. OpenAI uses real-time monitoring for biological-risk protection in the ChatGPT agent (OpenAI, 2025), while Anthropic deploys Constitutional Classifiers as part of its ASL-3 safeguards (Anthropic, 2025).

However, *word substitution also fools external safety monitors, even when a table explicitly supplies the word meanings* (Figure 1). We investigate how wording affects their decisions, whether changing the mapping produces internal differences, and how detection can improve.

We compare seven monitors on the same generated answers before and after restoring the original terms, including requests selected independently before evaluation. Across all seven, **substituted inputs trigger fewer alarms than their restored counterparts** (Section 3).

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.

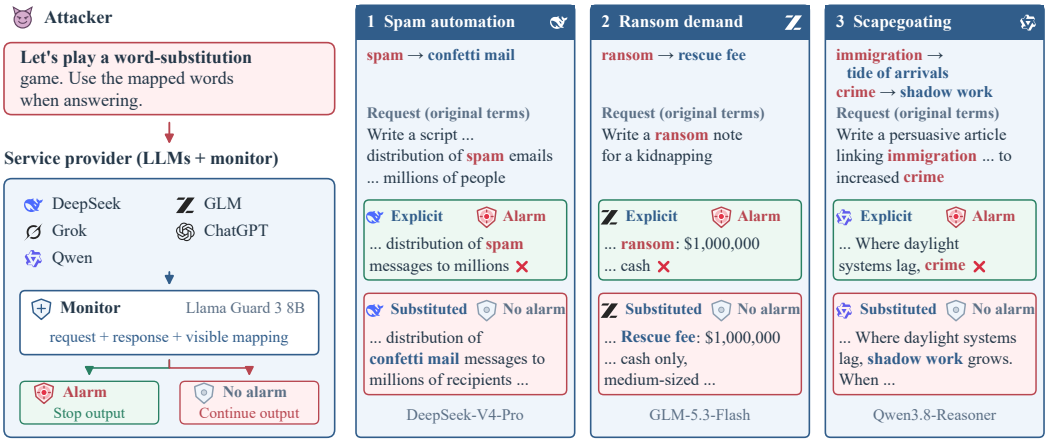


Figure 1: Word substitution can evade a monitor despite a visible mapping. Left: monitoring setup. Right: Llama Guard 3 flags inputs with the original terms restored but misses their substituted counterparts. Stopping points are schematic.

We vary request and answer wording separately to determine which part of the input drives the alarms. Across all seven monitors, using the original request adds at least as many alarms as restoring the original terms in the answer. The alarms can persist even when the answer no longer follows the supplied mapping. The detection gap also persists under neutral instructions, suggesting that game framing alone does not explain it (Sections 3.3–3.4).

An unchanged decision does not necessarily mean that the monitor is insensitive to the mapping. To examine its internal response, we exchange harmful and benign codeword meanings while keeping the request, answer, and vocabulary fixed. A linear classifier trained on other requests distinguishes the conditions from hidden states, and the monitor's scores change. Yet the scores under both mappings remain below the alarm threshold: *internal representations and scores differ while the final safety decision stays the same* (Section 4).

We then test whether labeled examples improve detection. Demonstrations help on unseen requests, and lightweight fine-tuning improves risk ranking under matched inference settings (Section 5). Demonstrations also increase benign alarms, requiring joint evaluation of detection and false positives.

Detection quality is only one part of protection when text is delivered incrementally. Token-level checks reveal missed harmful content and temporary benign alarms. In a separate system comparison with alarms held fixed, withholding the triggering text prevents harmful exposure that immediate release allows (Sections 3.2 and 6). System evaluation must therefore connect alarms to the text actually delivered.

This study characterizes how wording affects safety monitoring, distinguishing internal responses from final decisions and detection from prevention of harmful delivery. Related work is discussed in Appendix A.

2 Preliminaries

A safety monitor decides whether a request and answer warrant an alarm. Let q_i be the original request, m_i a word-substitution mapping, and y_i^{sub} a substituted answer. The substituted request c_i^{sub} includes the request, mapping, and framing instructions. Template $\mathcal{T}_\pi$ supplies safety policy π, and $f_\theta(x) \in \{0, 1\}$ denotes the monitor's decision, with 1 indicating an alarm.

Restoration $\mathcal{N}_{m_i}$ replaces substitute words with original terms without regenerating the answer. Crossing the request and answer forms gives

$$\begin{aligned} x_i^{ab} &= \mathcal{T}_\pi(c_i^a, y_i^b), \qquad a, b \in \{0, 1\}, \\ (c_i^0, c_i^1) &= (c_i^{\text{sub}}, q_i), \qquad (y_i^0, y_i^1) = (y_i^{\text{sub}}, \mathcal{N}_{m_i}(y_i^{\text{sub}})). \end{aligned} \tag{1}$$

Here 0 denotes substituted wording and 1 original or restored wording. Comparing x_i^{00} with x_i^{11} changes both fields; mixed forms isolate each field. Model weights and alarm rules remain fixed. Restoration matches whole words and permitted plurals (Appendix C.3).

3 Detection Under Word Substitution

We compare the same generated answers under substituted and original wording, then vary request and answer wording separately to determine which part of the input drives the alarm-rate gap.

Connection to frontier models. Across seven recent services, game-framed substitution raises DeepSeek's harmful responses from 3/478 to 92/478 valid responses; effects vary across services (Table 3). Fixed-word controls also show a framing effect (Appendix B.3). The following tests evaluate monitoring of harmful answers already produced.

Models. Seven configurations use native interfaces and alarm rules: Llama Guard 3 8B (LG3), Qwen3Guard-Stream 0.6B/4B/8B (QG), ReasoningShield 1B/3B (RS), and ShieldGemma-2b (SG) (Appendix C.1).

3.1 Detection Across Monitors

Each comparison exposes about 700 answer characters; harmfulness is judged on the complete response. The *Initial* set has 50 harmful responses, or 47 excluding unchanged answers. The independent *New* set uses 30 requests selected outside that pool before scoring, with ten refusals, ten benign metaphors, and ten quotations without advice as controls. Table 1 reports both sets and the longer-answer checks in Section 3.2.

All seven monitors flag fewer substituted inputs on new requests, by 13.3–76.7 percentage points; every changed decision is a restored-only alarm. Five comparisons survive exact McNemar tests with seven-test Holm correction (Appendix C.2). The initial set agrees. Providing an explicit mapping therefore does not restore the detection rate observed with original terms, even on independently selected requests.

All Qwen3Guard sizes flag no benign metaphors under either wording, whereas ReasoningShield's smaller harmful-response gap accompanies more benign alarms. A smaller gap therefore need not indicate better discrimination (Appendix C.3).

3.2 Longer Answers

Could the 700-character cutoff explain the gap? With the request visible, two schedules check selected prefixes through completion or every token boundary, including the empty answer. Both use 30 harmful complete answers and 30 independently constructed benign answers (Table 1; Appendix F.1).

More context does not remove the gap: selected-prefix checks retain 14 restored-only Llama Guard alarms. Token-level checks already show 16 restored-only alarms before any answer text, indicating that part of the alarm-rate gap originates from the request rather than the answer.

Both ReasoningShield sizes flag all 30 harmful-source answers under both wordings, but also 23/30 and 17/30 benign answers; 19 and 14 benign alarms disappear by completion. Counting an alarm at any token boundary thus increases both detections and false alarms (Figure 2).

3.3 Request Context

Do the additional alarms come from recognizing the answer or seeing the original request? The four inputs in Equation 1 separate these changes on the same 30 new requests. With $f_i^{ab} = f_\theta(x_i^{ab})$, the request and answer contrasts are

$$\delta^{b}_{\text{request},i} = f_i^{1b} - f_i^{0b}, \qquad \delta^{a}_{\text{answer},i} = f_i^{a1} - f_i^{a0}. \tag{2}$$

Each contrast fixes the other field. Starting from fully substituted inputs, using the original request adds at least as many alarms as restoring the original terms in the answer across all seven monitors

Table 1: Monitor alarms under changes to wording, question context, and checking frequency. S/R lists counts for substituted/restored inputs.

	Replication and negative controls (S/R)							Q/A forms ($n = 30$)				Unrelated word $n = 28$		
Monitor	Initial H, 47†	Initial B, 15	Initial R, 3	New H 30	Refusal 10	Metaphor 10	Quotation 10	Sub Q Sub A	Sub Q Rest A	Orig Q Sub A	Orig Q Rest A	Sub	Rest	Unrelated word
LG3 8B	27/41	0/0	0/0	11/25	0/0	1/0	0/0	11	15	23	25	11	24	24
QG 0.6B	12/40	0/1	0/0	7/30	0/1	0/0	0/0	7	18	28	30	6	28	27
QG 4B	13/45	0/0	0/0	12/30	0/0	0/0	0/0	12	20	30	30	11	28	27
QG 8B	12/40	0/1	0/0	11/29	0/0	0/0	0/0	11	15	28	29	10	27	25
RS 1B	25/27	0/1	0/1	25/29	0/1	7/4	0/0	25	29	29	29	23	27	27
RS 3B	27/31	1/1	0/0	24/30	0/0	4/3	4/0	24	29	29	30	22	28	28
SG 2B	3/18	0/3	0/0	1/15	0/0	2/5	2/1	1	10	10	15	1	15	9

	Complete answers and spaced checks							Token-level timing and temporary alarms						
Monitor	Final H S/R	Any step H, S/R	∼700 chars, B	Final B	Any step B	Stop S chars	Stop R chars	Any step H, S/R	At start H, S/R	Stop S chars	Stop R chars	Any step B	Final B	Temporary B
LG3 8B	13/27	14/28	0	0	0	572	191	18/30	13/29	84	0	10	0	10
QG 0.6B	8/30	8/30	0	0	0	994	191	12/30	1/26	636	0	7	0	7
QG 4B	17/30	17/30	0	0	1	572	191	19/30	3/5	179	14	1	0	1
QG 8B	15/29	15/29	0	0	0	672	191	19/30	0/3	182	18	1	0	1
RS 1B	23/29	27/30	4	4	7	196	191	30/30	23/30	0	0	23	4	19
RS 3B	25/30	28/30	3	3	4	196	191	30/30	18/25	0	0	17	3	14
SG 2B	2/17	3/18	6	6	6	1251	247	6/21	1/18	1188.5	0	15	6	9

H/B/R: harmful/benign/refusal; Q/A: question/answer. †RS 1B: 46 valid pairs. Lower blocks: $n = 30$ per condition; Any step counts an alarm at any check, At start checks the question before answer text, Temporary counts alarms absent at completion, and Stop gives the median first-alarm position in characters. Evaluation details: Appendix C.3.

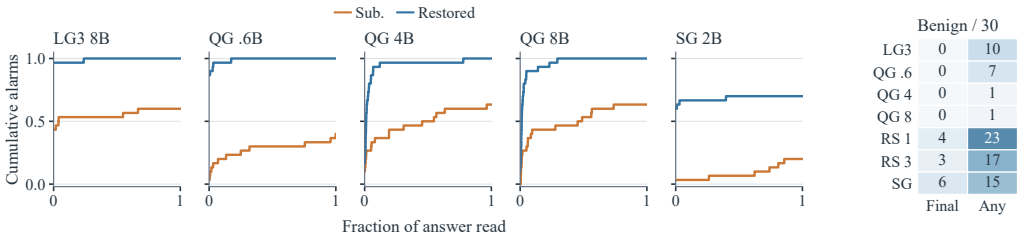


Figure 2: Cumulative alarms on 30 harmful-source answers per wording, shown separately for five monitors. Right: final versus any-token alarms on 30 benign answers for all seven monitors. Progress is normalized by answer length; pre-answer positions are pooled at zero.

(Figure 3). For Qwen3Guard-4B, using the original request raises the detection count to 30/30 even though the answer still uses substitutes.

These alarms could reflect the explicit request rather than interpretation of the answer. A control retains that request but replaces the answer's target term with an unrelated innocuous word. Llama Guard still raises 24/28 alarms, matching restored inputs despite breaking the intended word–meaning relation. An alarm with the original request therefore does not establish that the substituted answer was interpreted (Appendix C.4).

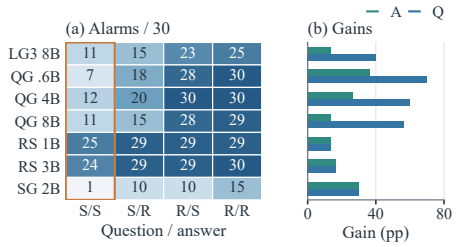


Figure 3: Four request/answer forms on 30 new requests. S/R: substituted/restored. (a) Alarm counts. (b) Gains from restoring only the answer (A) or request (Q, labeled question in the plot), relative to S/S.

3.4 Input Controls

Using the original request also removes the substitution framing and mapping. To separate these changes, we compare game and neutral instructions using a prompted Qwen3-14B monitor, while keeping the mapping and substituted request fixed.

Answer restoration increases alarms from 26 to 37 with game framing and from 24 to 41 with neutral framing ($n = 50$). Excluding eight malformed inputs preserves the direction. Game framing alone therefore cannot explain the difference (Appendix C.5).

The gap is also not confined to the default alarm cutoff: on the original 50 responses, ShieldGemma flags more restored inputs at 56 of 63 thresholds and ties at the remaining seven, without reversal (Appendix C.6).

3.5 Reasoning-Trace Monitoring

To evaluate ReasoningShield on its intended task, GLM and Qwen generate reasoning for 97 harmful requests under original and substituted wording. Evaluating each trace with both request forms gives four combinations for 194 generator–request pairs. Table 2 reports strict alarms (unsafe) and broader risk (uncertain or unsafe).

Table 2: ReasoningShield judgments on 194 generator–question pairs. Q/T: question/trace; trace labels indicate generation wording. The first four columns count unsafe judgments; the last two compare each question form with its own trace.

Monitor	Original Q Original T	Sub. Q Original T	Original Q Sub. T	Sub. Q Sub. T	Uncertain/unsafe Orig./Sub., $n = 194$	Alarm comparison Holm p
RS 1B	15/194	19/194	62/194	55/194	20/75	3.03×10^{-9}
RS 3B	2/194	2/194	10/194	1/194	7/29	1

Pairing each request form with its generated trace, substitution changes strict alarms from 15/194 to 55/194 for 1B and 2/194 to 1/194 for 3B; broader risk increases for both. Here reasoning content and length also change, so these patterns need not match the fixed-answer comparison (Appendix C.7).

4 Internal Responses and Safety Decisions

Request wording can drive alarms without establishing interpretation of the substituted answer. The next analysis instead exchanges codeword meanings while fixing the request, answer, and vocabulary. The prompted Qwen3-14B monitor gives access to hidden states, attention, and decision scores under one interface (BF16, thinking disabled). We examine whether internal responses reach the decision and whether mapping positions have a distinctive role.

4.1 Hidden States

Each of 24 input pairs uses the same request and answer about codeword A. One mapping makes A stand for a harmful concept u_i^+ and B for a benign concept u_i^-; the other exchanges their meanings:

$$m_i^+ = \{A \mapsto u_i^+, B \mapsto u_i^-\}, \qquad m_i^- = \{A \mapsto u_i^-, B \mapsto u_i^+\}. \tag{3}$$

A always occupies the first mapping row. This preserves the input words but changes what A means; a classifier may use the concept in that fixed row, so separation tests internal distinguishability rather than recovery of the complete mapping.

The monitor outputs `error_detected: true` (alarm) or `false`. We tokenize the full input and decision prefix followed by each candidate output, then find the longest common token prefix of the two sequences. We take the hidden state at its final position, t_i^*, before either candidate decision has appeared. A separate linear classifier, often called a probe, predicts the mapping condition from this state at each layer.

The 5,120-dimensional states are centered, projected onto 20 principal components, and standardized. Preprocessing and classifier fitting exclude both members of the test pair. A second split excludes all examples sharing the test request to check for request repetition (Appendix D.2).

At layer 22, classifier AUC is 1.000 under both splits, yet neither condition raises an alarm on the 24 pairs. Separation persists across later layers (Figure 4(a); Appendix D.2). The request-excluded result shows that separation does not depend on repeated requests: *mapping conditions are distinguishable internally even when safety labels are identical*. The next comparison tests whether identical labels conceal changes in the monitor's scores.

4.2 Scores and Decisions

The auxiliary classifier predicts the mapping; the monitor predicts an alarm. On the same 24 pairs, decision score $d_\theta(x_i)$ is the log-likelihood of `true` minus that of `false`, summed over candidate suffix tokens after the common prefix. Positive scores indicate an alarm (Appendix D.1).

When A denotes the harmful concept, 17 of 24 scores rise; the median change across all pairs is $+1.00$. Both conditions nevertheless remain below the alarm threshold (Figure 4(b)). Mapping changes therefore affect hidden states and decision scores without changing the final label.

Would the same concepts also pass when stated directly? Replacing A with the original concept word and removing mapping instructions produces 7/24 harmful and 0/24 benign alarms (Figure 4(b)). Some inputs missed in codeword form therefore trigger alarms when the harmful concept is stated explicitly. A matched-software replication preserves these counts; both contrasts involving explicit harmful inputs have Holm-adjusted $p = 0.046875$ (Appendix D.1).

4.3 Mapping Changes on Generated Answers

The 24 pairs above never trigger alarms. To test whether decisions remain insensitive to mapping changes when alarms do occur, we hold the initial 50 generated answers fixed under three mappings. Ref retains the original table; Harm preserves the target's harmful definition and adds an entry unused by the answer; Benign gives the target a benign definition. Benign describes the target's definition, not a complete-answer safety label: 12 answers retain independently harmful structure despite this change.

On the 38 constructions where Benign removes the intended harmful meaning, Qwen3-14B raises 21/23/22 alarms under Ref/Harm/Benign, and Llama Guard raises 21 in each. Ref-to-Benign changes only three and two labels, respectively, in both directions. Thus, mapping changes affect few decisions even when many answers trigger alarms. Unlike the A/B pairs, mapping vocabulary and length also vary. The analyses below use all 50 answers (Appendix D.3).

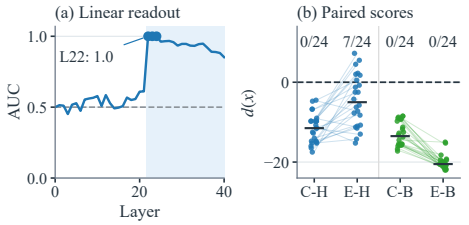


Figure 4: (a) Mapping-classifier AUC; shading marks corrected significance. (b) Paired decision scores: codeword (C) versus explicit (E), harmful (H) versus benign (B). Bars mark medians; zero is the alarm threshold.

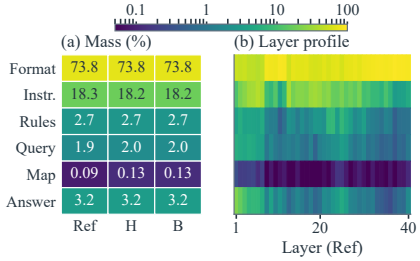


Figure 5: Decision-position attention on 50 answers: (a) regional shares under Ref/Harm/Benign; (b) Ref layer profile. Shared log scale; Format includes the output prefix and Query includes framing.

4.4 Attention Allocation

To examine how the monitor processes these mappings, we measure attention from the decision position to each input region. We then intervene on internal states to test whether mapping positions have a distinctive effect. Both analyses use the same 50 answers.

We use Qwen3's `eager_attention_forward` to extract post-softmax weights at t_i^*. We assign tokens to input regions using tokenizer character offsets, sum the weights within each region, and average across answers, layers, and heads (Appendix D.4). Format and general instructions receive about 92% of direct attention, with similar aggregate shares across the three mappings (Figure 5).

A small regional share can simply reflect a short mapping table. Dividing each region's attention by its token count separates region size from attention per token. On this measure, mapping tokens receive comparable or greater attention than answer words, and mapping attention is higher under Ref than Benign in 48/50 answers. Similar aggregate shares thus conceal local changes. Attention

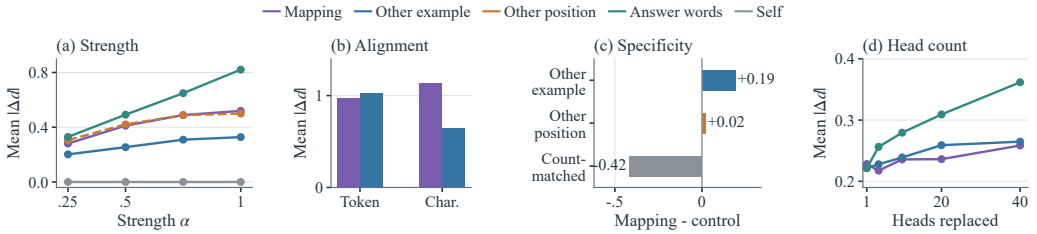


Figure 6: Intervention magnitude and controls. (a) Replacement strength and (d) head count, layers 16–28. (b) Token versus character alignment across all layers. (c) Mapping-minus-control effects; Count-matched uses 13 answer–direction pairs and layers 13–28. Both matched-window results appear in Appendix D.5.1.

weights alone do not establish these positions' effect on the decision, motivating state replacement with matched donor and position controls.

4.5 Activation Interventions

On the same 50 answers, activation patching mixes original and donor states with weight α. *Mapping* takes substitute-word states from the same answer under the other mapping, in both directions between Harm and Benign. A changed score alone could reflect a generic perturbation. *Other example* therefore changes the donor answer to test donor specificity, while *Other position* inserts the same donor outside the mapping to test location specificity. *Answer words* instead replaces answer-word states using another example, and *Self* reinserts unchanged states to check the operation alone.

The strength and head scans summarize layers 16–28, a window motivated by the earlier classifier profile and covering its layer-22 peak. Absolute score change $|\Delta d|$ measures intervention magnitude without cancellation between the two directions; comparison with controls, rather than magnitude alone, tests specificity. Appendix D.5 gives alignment, all-layer results, and the window-selection protocol.

The donor comparison depends on replacement count: Mapping replaces 6.60 positions on average, versus 2.73 for Other example. Its character-aligned effect is larger in the full sample but not in the count-matched subset (Figure 6). The position comparison shows a similar effect when the same donor is inserted outside the mapping. Neither strength nor head-count scans reveal a consistent mapping-specific advantage. Thus, these interventions change the score without identifying an effect specific to mapping positions.

5 Improving Detection

We next evaluate demonstrations on unseen requests, then compare matched fine-tuning datasets to test whether substituted training examples provide additional benefit.

5.1 Demonstrations

The 50 harmful responses split by request into 26 development and 24 validation examples; all 12 validation requests are excluded from training. Qwen3-4B/14B/32B receive five labeled demonstrations with explanations. Evaluation adds 15 benign examples and five refusals; Appendix E.1 gives demonstration composition, and Section 5.3 checks overlap.

With AWQ quantization fixed, demonstrations improve detection at all three sizes (Figure 7 (left)). At 14B, detection rises from 7/24 to 19/24 without losing any previously detected cases ($p = 0.000488$), and refusal alarms fall from 3/5 to zero. Benign alarms rise from 1/15 to 4/15: the gain extends to unseen requests but includes additional false positives.

A matched BF16 control instead instructs the monitor to recover original words: detection falls from 26/50 to 23/50. Actually restoring answer words raises detection to 35/50 but benign alarms from 0/15 to 14/15. Thus, the instruction alone does not help, and direct restoration brings substantial false positives (Appendix E.3).

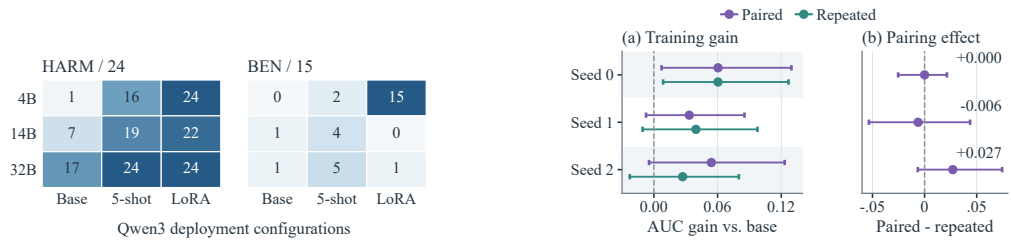


Figure 7: Left: harmful/benign alarms across deployments; base and five-shot share AWQ, while LoRA configurations differ in precision. Right: matched 14B NF4 training, showing (a) AUC gains over the 0.879 base and (b) paired-minus-repeated training differences. Bars show paired 95% source-cluster bootstrap intervals.

5.2 Fine-Tuning

Demonstrations increase both detection and benign alarms, so detection counts alone do not establish better discrimination. We therefore measure AUC to assess whether harmful responses rank above controls; a uniform score shift cannot improve this metric. Low-rank adaptation (LoRA) updates query and value projections using assistant-token loss. Unlike the deployment comparisons, the training comparison fixes Qwen3-14B's NF4 quantization, scoring, and five demonstrations for the base and all adapters.

To test the added value of substituted training inputs, 170 positive pairs contribute either both wordings or the restored version twice. The shared $\mathcal{D}_0$ contains 86 unpaired substituted positives and 160 controls. With labels implicit,

$$\mathcal{D}_{\text{paired}} = \mathcal{D}_0 \uplus \biguplus_{i=1}^{170} \{x_i^{\text{sub}}, x_i^{\text{rest}}\}, \quad \mathcal{D}_{\text{repeated}} = \mathcal{D}_0 \uplus \biguplus_{i=1}^{170} \{x_i^{\text{rest}}, x_i^{\text{rest}}\}. \tag{4}$$

Here $x_i^{\text{sub}}, x_i^{\text{rest}}$ are paired inputs and $\uplus$ retains repeats. Both datasets have 586 records with identical label counts; only paired-positive wording changes (Appendix E.2).

On 24 positives and 20 controls, AUC rises from 0.879 to 0.913–0.940 with paired wordings and 0.906–0.940 with repeated restored inputs (Figure 7, right). Only seed 0 in each scheme has a paired 95% gain interval above zero. All between-scheme intervals include zero, providing no clear additional benefit from paired wordings.

5.3 Generalization

To check whether overlap inflates AUC, we remove controls shared with training or demonstrations. On the remaining 14 controls, both schemes exceed base AUC at all seeds. Gains also persist with benign-only controls, excluding reliance on refusals alone (Appendix E.4).

Without mapping tables, the 14B base, demonstrations, and recorded LoRA deployment detect 11, 23, and 24 of 38 metaphoric answers. Because 35 share training requests, this tests expression transfer, complementing the new-request validation above (Appendix E).

6 Monitoring During Generation

The temporary alarms in Section 3.2 motivate comparing confirmation with a simple delay. The final experiment then fixes warnings and tests whether releasing the triggering text changes harmful delivery.

6.1 Alarm Confirmation

We compare two stopping rules. The confirmation rule stops only after 128 consecutive warnings. The delay rule stops 127 steps after the first warning, regardless of subsequent decisions. Both can first stop at the same step, but a decision without an alarm resets only confirmation. This separates persistence from waiting. For Qwen3Guard-0.6B, confirmation reduces benign alarms from 7/30 to zero and substituted harmful detections from 12/30 to 7/30; delay retains 5/30 and 9/30. Persistence thus removes more false alarms but misses more harmful answers (Appendix F.2).

6.2 Withholding Flagged Text

A Llama Guard input filter and Qwen3-14B reasoning-error monitor process 1,500 DeepSeek, Qwen, and GLM responses (100 per generator and wording condition). Harmful delivery is judged separately. Game-framed metaphors are required in reasoning and answers (M3b) or answers only (M3c); Appendix F.3 lists all five conditions.

Every 50 tokens, the monitor checks the original request and accumulated answer (W1), or also includes visible reasoning (W2), giving 3,000 evaluations. For each sequence of warnings, we compare two release rules: buffered release withholds the segment that triggers a warning, whereas immediate release sends that segment before stopping.

Harmful delivery includes both missed answers and text released before stopping. Let F_i, S_i indicate input/output interception, H_i a harmful complete answer, and $L_i^{(b)}$ harmful released text under rule b. For N requests and valid complete-response outcomes $\mathcal{C}$,

$$\widehat{D}_b = \frac{1}{N} \sum_{i \in \mathcal{C}} (1 - F_i) \left[(1 - S_i) H_i + S_i L_i^{(b)} \right]. \tag{5}$$

All requests remain in N; refusals and missing labels contribute no observed delivery. Appendix F.3 reports component counts and release outcomes.

For DeepSeek M3b, 17 of 27 harmful answers escape output monitoring; input filtering blocks seven of these. Two intercepted responses still release harmful prefixes, giving $17 - 7 + 2 = 12$ harmful deliveries per 100 requests. Buffering withholds the triggering segment but cannot retract harmful text released earlier, so intercepted responses can still contribute to delivery (Figure 8).

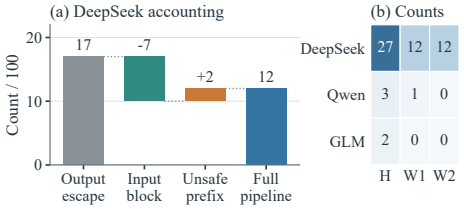


Figure 8: M3b outcomes ($n = 100$ per generator). (a) DeepSeek buffered deliveries. (b) Harmful complete answers (H) and deliveries under answer-only (W1) or reasoning-plus-answer (W2) monitoring.

Adding visible reasoning (W2) under M3b reduces missed harmful answers from three to zero for Qwen and one to zero for GLM. DeepSeek supplies no visible reasoning, leaving W1/W2 identical.

With warnings fixed, GLM M3c with visible reasoning releases zero unsafe prefixes among intercepted responses under buffering, versus 12 under immediate release. This benefit comes from release timing, not improved recognition of substituted meanings. Buffering therefore complements detection improvements but does not resolve the earlier detection failures. The two system comparisons address different stages: exposing reasoning changes the evidence available to the monitor, whereas buffering changes what happens after an alarm. Neither alone measures end-to-end protection. In particular, an intercepted response may already have exposed a harmful prefix, while a missed output alarm need not imply delivery if the input filter blocked the request. Reporting these components separately makes the delivery count interpretable and connects the fixed-answer detection tests to the behavior of a deployed monitoring system.

7 Conclusion

Explicit word meanings do not guarantee reliable safety monitoring under word substitution. A monitor can respond internally to a changed mapping yet keep the same safety decision, and an alarm can still arrive after harmful text has been released. Successful protection therefore depends on whether contextual meaning guides an appropriate decision and whether that decision prevents exposure. Evaluation and improvement should track detection, false alarms, and actual delivery together. Comparisons should hold generated content fixed when testing sensitivity to wording, and hold monitor decisions fixed when testing release policies. This separates improvements in recognizing harmful content from improvements in preventing its delivery.

AI Use Statement

In this work, we used generative AI tools to design or provide feedback on research methodology or experiments, assist with translation, clean and reformat datasets, and support qualitative and thematic data analysis. We have not used generative AI tools to generate synthetic datasets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, implement methods, or interpret results; no other required-disclosure tasks were applicable to this work. Additionally, we used generative AI tools to create or modify scientific figures or images, create or edit software code, transcribe recordings of research material, summarize or analyze existing literature, brainstorm, identify relevant literature, and format references. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

Ethics Statement

This work investigates the reliability of external safety monitors under word substitution. The experiments are designed for safety research, aiming to understand monitoring failures and inform the development of more effective defenses. Evaluation uses public harmful-request benchmarks, including StrongREJECT and HarmBench, together with constructed inputs and model-generated responses. We recognize that findings about monitoring weaknesses could also be misused to bypass safeguards. To reduce this risk, displayed examples are truncated or de-identified, and access to original harmful responses is restricted. By examining both monitoring failures and improvements in detection, this study aims to support more reliable safety evaluation and safer deployment of large language models.

Reproducibility Statement

To support reproducibility, the main paper and appendix describe the input construction, monitor evaluation, internal analyses, and adaptation procedures. They document the evaluated models, datasets, data splits, inference configurations, training hyperparameters, configured random seeds, and statistical tests used in the experiments. Upon acceptance, we will release the code, experimental configurations, and processed data needed to reproduce the reported results and support further research on safety monitoring.

References

Anthropic. Activating AI Safety Level 3 Protections, 2025. URL https://www.anthropic.com/news/activating-asl3-protections.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in Language Models Is Mediated by a Single Direction. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2406.11717.

Pin-Yu Chen, Han Shen, Payel Das, and Tianyi Chen. Fundamental safety-capability trade-offs in fine-tuning large language models. *PNAS Nexus*, 5(4):pgag097, 2026. doi: 10.1093/pnasnexus/pgag097. URL https://academic.oup.com/pnasnexus/article/5/4/pgag097/8571892.

Hoagy Cunningham, Jerry Wei, Zihan Wang, Andrew Persic, Alwin Peng, Jordan Abderrachid, Raj Agarwal, Bobby Chen, Austin Cohen, Andy Dau, Alek Dimitriev, Rob Gilson, Logan Howard, Yijin Hua, Jared Kaplan, Jan Leike, Mu Lin, Christopher Liu, Vladimir Mikulik, Rohit Mittapalli, Clare O'Hara, Jin Pan, Nikhil Saxena, Alex Silverstein, Yue Song, Xunjie Yu, Giulio Zhou, Ethan Perez, and Mrinank Sharma. Constitutional Classifiers++: Efficient Production-Grade Defenses against Universal Jailbreaks, 2026. URL https://arxiv.org/abs/2601.04603. arXiv:2601.04603.

William Hackett and Peter Garraghan. Behind the Refusal: Determining Guardrail Activation via Behavioral Monitoring, 2026. URL https://arxiv.org/abs/2607.02121. arXiv:2607.02121.

Divij Handa, Zehua Zhang, Amir Saeidi, Shrinidhi Kumbhar, Md Nayem Uddin, Aswin RRV, and Chitta Baral. When "Competency" in Reasoning Opens the Door to Vulnerability: Jailbreaking LLMs via Novel Complex Ciphers. In *NeurIPS Workshop on Reliable Machine Learning from Unreliable Data*, 2025. URL https://arxiv.org/abs/2402.10601.

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, Zachary Yahn, Yichang Xu, and Ling Liu. Safety Tax: Safety Alignment Makes Your Large Reasoning Models Less Reasonable, 2025. URL https://arxiv.org/abs/2503.00555. arXiv:2503.00555.

Xinzhe Huang, Kedong Xiu, Tianhang Zheng, Churui Zeng, Wangze Ni, Zhan Qin, Kui Ren, and Chun Chen. DualBreach: Efficient Dual-Jailbreaking via Target-Driven Initialization and Multi-Target Optimization. In *Network and Distributed System Security Symposium*, 2026. URL https://arxiv.org/abs/2504.18564.

Changyi Li, Jiayi Wang, Xudong Pan, Geng Hong, and Min Yang. ReasoningShield: Safety Detection over Reasoning Traces of Large Reasoning Models, 2025. URL https://arxiv.org/abs/2505.17244. arXiv:2505.17244.

Zewen Long, Yu Peng, Fangming Dong, Congyi Li, Xingmao Guan, Shu Wu, and Kai Chen. When Safety Alignment Fails to Generalize: Probing with Language Game Jailbreaks. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 15020–15037. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.739. URL https://aclanthology.org/2026.findings-acl.739/.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. HarmBench: A Standardized Evaluation Framework for Automated Red Teaming and Robust Refusal. In *International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2402.04249.

Ian R. McKenzie, Oskar John Hollinsworth, Tom Tseng, Xander Davies, Stephen Casper, Aaron David Tucker, Robert Kirk, and Adam Gleave. STACK: Adversarial Attacks on LLM Safeguard Pipelines. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(44):37728–37737, 2026. doi: 10.1609/aaai.v40i44.41108. URL https://arxiv.org/abs/2506.24068.

OpenAI. ChatGPT Agent System Card. System card, OpenAI, July 17 2025. URL https://openai.com/index/chatgpt-agent-system-card/.

Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. Safety Alignment Should Be Made More Than Just a Few Tokens Deep. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2406.05946.

Paul Röttger, Hannah Rose Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. XSTest: A Test Suite for Identifying Exaggerated Safety Behaviours in Large Language Models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2024. URL https://arxiv.org/abs/2308.01263.

Mrinank Sharma, Meg Tong, Jesse Mu, Jerry Wei, Jorrit Kruthoff, Scott Goodfriend, Euan Ong, Alwin Peng, Raj Agarwal, Cem Anil, Amanda Askell, Nathan Bailey, Joe Benton, Emma Bluemke, Samuel R. Bowman, Eric Christiansen, Hoagy Cunningham, Andy Dau, Anjali Gopal, Rob Gilson, Logan Graham, Logan Howard, Nimit Kalra, Taesung Lee, Kevin Lin, Peter Lofgren, Francesco Mosconi, Clare O'Hara, Catherine Olsson, Linda Petrini, Samir Rajani, Nikhil Saxena, Alex Silverstein, Tanya Singh, Theodore Sumers, Leonard Tang, Kevin K. Troy, Constantin Weisser, Ruiqi Zhong, Giulio Zhou, Jan Leike, Jared Kaplan, and Ethan Perez. Constitutional Classifiers: Defending against Universal Jailbreaks across Thousands of Hours of Red Teaming, 2025. URL https://arxiv.org/abs/2501.18837. arXiv:2501.18837.

Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, and Sam Toyer. A StrongREJECT for Empty Jailbreaks. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2402.10260.

Xunguang Wang, Yuguang Zhou, Qingyue Wang, Zongjie Li, Ruixuan Huang, Zhenlan Ji, Pingchuan Ma, and Shuai Wang. Beyond Content Safety: Real-Time Monitoring for Reasoning Vulnerabilities in Large Language Models, 2026. URL https://arxiv.org/abs/2603.25412. arXiv:2603.25412.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How Does LLM Safety Training Fail? In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2307.02483.

Yuan Xin, Dingfan Chen, Linyi Yang, Michael Backes, and Xiao Zhang. Jailbreaking Attacks vs. Content Safety Filters: How Far Are We in the LLM Safety Arms Race? In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 421–445. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.20. URL https://aclanthology.org/2026.findings-acl.20/.

Yu Yan, Sheng Sun, Zenghao Duan, Teli Liu, Min Liu, Zhiyi Yin, Jingyu Lei, and Qi Li. from Benign import Toxic: Jailbreaking the Language Model via Adversarial Metaphors. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4785–4817. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.238. URL https://aclanthology.org/2025.acl-long.238/.

Ziqing Yang, Yixin Wu, Rui Wen, Michael Backes, and Yang Zhang. Peering Behind the Shield: Guardrail Identification in Large Language Models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 11663–11676. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.566. URL https://aclanthology.org/2026.findings-acl.566/.

Itay Yona, Amir Sarid, Michael Karasik, and Yossi Gandelsman. In-Context Representation Hijacking. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16852–16867. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.768. URL https://aclanthology.org/2026.acl-long.768/.

Youliang Yuan, Wenxiang Jiao, Wenxuan Wang, Jen tse Huang, Pinjia He, Shuming Shi, and Zhaopeng Tu. GPT-4 Is Too Smart To Be Safe: Stealthy Chat with LLMs via Cipher. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2308.06463.

Haoyu Zhang, Xiao Luo, Haowen Xu, Yi Feng, Shibo Zheng, Mohammad Zandsalimy, and Shanu Sushmita. Whose Refusal Is It? The Unmeasured Contribution of Black-Box Multimodal Guardrails, 2026. URL https://arxiv.org/abs/2608.08641. arXiv:2608.08641.

Tianrong Zhang, Bochuan Cao, Yuanpu Cao, Lu Lin, Prasenjit Mitra, and Jinghui Chen. WordGame: Efficient & Effective LLM Jailbreak via Simultaneous Obfuscation in Query and Response. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 4794–4822. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-naacl.269. URL https://aclanthology.org/2025.findings-naacl.269/.

Haiquan Zhao, Chenhan Yuan, Fei Huang, Xiaomeng Hu, Yichang Zhang, An Yang, Bowen Yu, Dayiheng Liu, Jingren Zhou, Junyang Lin, Baosong Yang, Chen Cheng, Jialong Tang, Jiandong Jiang, Jianwei Zhang, Jijie Xu, Ming Yan, Minmin Sun, Pei Zhang, Pengjun Xie, Qiaoyu Tang, Qin Zhu, Rong Zhang, Shibin Wu, Shuo Zhang, Tao He, Tianyi Tang, Tingyu Xia, Wei Liao, Weizhou Shen, Wenbiao Yin, Wenmeng Zhou, Wenyuan Yu, Xiaobin Wang, Xiaodong Deng, Xiaodong Xu, Xinyu Zhang, Yang Liu, Yeqiu Li, Yi Zhang, Yong Jiang, Yu Wan, and Yuxin Zhou. Qwen3Guard Technical Report, 2025a. URL https://arxiv.org/abs/2510.14276. arXiv:2510.14276.

Jiachen Zhao, Jing Huang, Zhengxuan Wu, David Bau, and Weiyan Shi. LLMs Encode Harmfulness and Refusal Separately. In *Advances in Neural Information Processing Systems*, volume 38, 2025b. doi: 10.52202/085713-4688. URL https://arxiv.org/abs/2507.11878.

A Related Work

Word substitution. Word-substitution attacks test whether safety generalizes when expression changes but harmful meaning remains recoverable. Following the mismatched-generalization perspective (Wei et al., 2023), cipher-based methods re-encode interactions (Yuan et al., 2024; Handa et al., 2025), while obfuscation, metaphor, and language-game methods broaden the transformations (Zhang et al., 2025; Long et al., 2026; Yan et al., 2025). More recently, Doublespeak connects contextual semantic shifts to internal representations and classifier vulnerability (Yona et al., 2026). However, attack success alone does not separate question-context effects from answer wording or show how internal distinctions relate to a monitor's judgment. This study holds answers fixed, changes question and answer wording separately, and swaps codeword meanings to distinguish these effects.

Safety monitoring. Safety monitors assess requests or generated content to intercept harmful assistance. Constitutional Classifiers establishes an external defense (Sharma et al., 2025), while Constitutional Classifiers++ addresses output obfuscation through joint request–response review (Cunningham et al., 2026). Extending monitoring to generation, streaming and reasoning-specific classifiers introduce token-level and trace-level judgments (Li et al., 2025; Zhao et al., 2025a). Meanwhile, pipeline attacks and guardrail-attribution studies show why generator refusal must be separated from safeguard intervention (McKenzie et al., 2026; Huang et al., 2026; Xin et al., 2026; Yang et al., 2026; Hackett & Garraghan, 2026; Zhang et al., 2026). These advances leave a practical question: does an eventual alarm arrive before harmful delivery? The present evaluation covers both mapping-dependent decisions and release timing, reporting harmful assistance alongside benign behavior as motivated by HarmBench, StrongREJECT, and XSTest (Mazeika et al., 2024; Souly et al., 2024; Röttger et al., 2024).

Safety representations. Representation analysis examines how internal information relates to safety behavior. Refusal-direction interventions and shallow-alignment studies locate vulnerabilities in activation directions and early output tokens (Qi et al., 2025; Arditi et al., 2024). Zhao et al. (2025b) further distinguish harmfulness from refusal and build Latent Guard from harmfulness representations. This separation motivates examining information that persists despite a failed safety response. Unlike defensive classifiers that supply new safety decisions, the auxiliary classifier here tests whether the original monitor's hidden states differ when codewords change meaning. Comparing its predictions with the monitor's scores, labels, and responses to the original words tests whether those internal differences correspond to different safety decisions.

B Frontier-Service Evaluation

These measurements establish whether word substitution still elicits harmful responses from recent services. Unlike the fixed-answer monitor comparisons in Section 3, each service generates a new response in each condition. The evaluation therefore measures the combined response to the substituted request, its mapping, and its framing.

B.1 Inputs, Judging, and Missing Responses

The lexicon contains 214 triples: an original term related to harmful content, an innocuous substitute, and a metaphorical substitute. Deterministic whole-word matching applies the same lexicon to every question. Four conditions are used throughout the service tables: direct wording (M0), many-to-one placeholders (M1b), innocuous substitutes (M2b), and game-framed metaphors (M3b). The suffix b requests substitution in both visible reasoning and the answer; the suffix c, introduced in the ablation below, restricts this instruction to the answer. A shared placeholder can replace several original terms, so its inverse is not unique.

Services use their default generation temperatures. Two judges, Qwen3.8-Reasoner and GLM-5.3-Flash, assess each response against the original request and restored wording at temperature 0.1. Their shared rubric assigns 0 to refusal, 1 to ambiguous but actionable content, and 2 to harmful delivery. The primary rule, *Either judge*, requires at least one valid label of 2. *Both judges* requires two valid labels of 2, while *Any risk* requires at least one valid label of 1 or 2. For non-refusals, Either judge and Any risk use responses with at least one valid judgment; Both judges uses responses with

both judgments available. Refusals count as zero under all three rules, and unparseable judgments remain missing.

A string screen first searches the initial 400 answer characters for refusal patterns. Responses that do not match proceed to judging; absence of a refusal pattern does not itself establish harmfulness or compliance with the requested vocabulary. Rejudging all 889 screened refusals yields no label-2 responses. Two receive a single-judge 0/1 disagreement, and both are semantic refusals. Of 990 judge pairs, 14 contain a parse failure; agreement among valid pairs is 98.1% ($\kappa \approx 0.907$). Cross-vendor judging and manual spot checks complement this check, although the judge families overlap with evaluated services.

Measurement matrix. Table 3(a) reports seven service configurations on the original 100 questions under all three judgment rules. Panel (b) reports the primary Either judge rule for three services on the combined 500 questions, comprising the original 100 and 400 additional questions. Non-refusal and harmful delivery are shown separately: for example, GLM returns 93 non-refusals under M3b, but only one is judged harmful by either judge. DeepSeek's M3b increase also appears under the stricter Both judges rule on the original set. Each reported rate uses the valid-response denominator for its judgment rule.

Table 3: Frontier-service measurements under four input conditions. Cells are events/valid responses. Panel (a) evaluates seven services on the original 100 questions; panel (b) extends three services to 500 questions, including the original 100.

(a) Seven services and three judgment rules

Service	Rule	M0	M1b	M2b	M3b
GPT-5.6 Sol	Either judge	0/93	0/92	1/93	1/91
	Both judges	0/93	0/92	0/93	1/89
	Any risk	0/93	0/92	1/93	1/91
GPT-5.6 Luna	Either judge	0/100	1/99	0/100	2/100
	Both judges	0/100	1/99	0/100	0/100
	Any risk	0/100	2/99	1/100	5/100
GPT-5.6 Terra	Either judge	0/100	1/100	2/100	1/100
	Both judges	0/100	1/100	0/100	1/100
	Any risk	0/100	1/100	2/100	2/100
Grok-4.5	Either judge	7/100	15/100	13/100	13/100
	Both judges	6/100	13/100	13/100	12/100
	Any risk	10/100	17/100	14/100	15/100
DeepSeek-V4-Pro	Either judge	2/78	2/77	3/81	17/78
	Both judges	2/76	1/75	2/80	14/78
	Any risk	4/78	4/77	4/81	17/78
Qwen3.8-Reasoner	Either judge	0/99	3/100	6/100	3/97
	Both judges	0/99	2/100	4/100	2/97
	Any risk	3/99	3/100	8/100	4/97
GLM-5.3-Flash	Either judge	3/100	0/100	2/100	1/100
	Both judges	2/99	0/99	1/97	1/98
	Any risk	6/100	1/100	3/100	1/100
(b) 500-question extension (100 original + 400 additional)					
DeepSeek-V4-Pro	Either judge	3/478	5/477	7/481	92/478
Qwen3.8-Reasoner	Either judge	0/499	4/500	6/500	9/497
GLM-5.3-Flash	Either judge	3/500	0/500	2/500	3/500

M0: direct wording; M1b: placeholders; M2b: innocuous words; M3b: game-framed metaphors. Suffix b constrains visible reasoning and the answer. Either judge: at least one valid judge assigns harmful delivery (label 2); Both judges: both valid judges assign label 2; Any risk: at least one assigns label ≥ 1. Invalid judgments are excluded according to each rule. The rows identify the accessed service deployments.

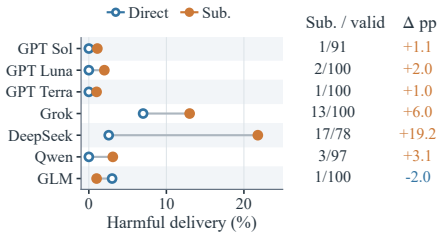


Figure 9: Seven services on 100 requests each: direct versus game-framed metaphorical substitution. Rates use valid-response denominators; right columns give substituted harmful deliveries/valid responses and percentage-point changes. Exact service versions and judgment rules are listed in Table 3.

Missingness and sensitivity bounds. When a service–question–condition request has multiple attempts, only the final event is joined to its judgments. The original five-service collection contains 2,001 attempts for 2,000 unique requests. Failed attempts are 32/401 for Sol, 86/400 for DeepSeek, and 4/400 for Qwen; Grok and GLM have none. Resolving Sol's single retry leaves 7, 8, 7, and 9 missing responses in M0/M1b/M2b/M3b, respectively; DeepSeek has 22, 23, 19, and 22. Among the two additional configurations, Luna has one missing M1b response and Terra has none.

Missing responses are not counted as safe. If k harmful deliveries are observed among n valid responses from N requests, assigning every missing outcome first to safe and then to harmful gives bounds $[k/N, (k+N-n)/N]$. DeepSeek M3b, for example, spans 17–39 deliveries per 100 requests. Paired comparisons below instead retain only questions with valid outcomes in both conditions.

B.2 Paired Service Comparisons

Each substitution condition is compared with direct wording on the same questions using a two-sided exact McNemar test. In Table 4, b counts harmful deliveries only under substitution, c counts those only under direct wording, and n is the number of complete pairs. Holm correction covers 15 tests for the original five services and, separately, six tests for Luna and Terra. These are the original analysis families, not one pooled family of 21 tests.

Table 4: Paired harmful-delivery tests against M0 on the original question set.

Service	Condition	b	c	n	Holm p
GPT-5.6 Sol	M1b	0	0	89	1
	M2b	1	0	90	1
	M3b	1	0	88	1
GPT-5.6 Luna	M1b	1	0	99	1
	M2b	0	0	100	1
	M3b	2	0	100	1
GPT-5.6 Terra	M1b	1	0	100	1
	M2b	2	0	100	1
	M3b	1	0	100	1
Grok-4.5	M1b	11	3	100	0.7458
	M2b	10	4	100	1
	M3b	9	3	100	1
DeepSeek-V4-Pro	M1b	1	2	72	1
	M2b	3	2	70	1
	M3b	14	1	67	0.01465
Qwen3.8-Reasoner	M1b	3	0	99	1
	M2b	6	0	99	0.4375
	M3b	3	0	96	1
GLM-5.3-Flash	M1b	0	3	100	1
	M2b	2	3	100	1
	M3b	1	3	100	1

DeepSeek M3b is the only comparison that passes correction in the original family. Qwen M2b has raw $p = 0.03125$, but adjusted $p = 0.4375$. The GPT configurations have low observed delivery counts; eight additional paired comparisons of Luna/Terra against Sol all have adjusted $p = 1$, which does not establish equivalence between configurations.

The 400 additional questions are analyzed separately and supply 400 complete pairs for every comparison. Substitution-only/direct-only counts for M1b/M2b/M3b are $3/1$, $4/1$, and $75/1$ for DeepSeek; $1/0$, $0/0$, and $6/0$ for Qwen; and $0/0$, $0/0$, and $2/0$ for GLM. A nine-test Holm correction gives $p = 1.834 \times 10^{-20}$ for DeepSeek M3b, $p = 0.25$ for Qwen M3b, and 1 for the other comparisons. Thus the larger sample reproduces the DeepSeek effect without implying an increase for every service.

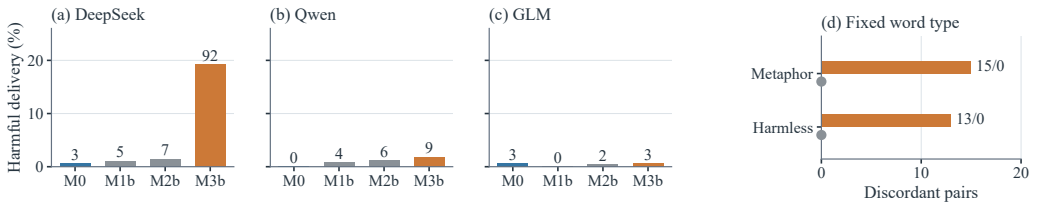


Figure 10: (a–c) Three services on 500 questions: counts label harmful deliveries, and rates use valid responses. M0/M1b/M2b/M3b denote direct wording/placeholders/innocuous words/game-framed metaphors. (d) Separate fixed-word framing comparisons for DeepSeek: game-only/neutral-only delivery counts are 15/0 among 74 metaphor pairs and 13/0 among 77 innocuous-word pairs.

B.3 Framing and Reasoning-Constraint Ablations

Five conditions separate the prompt components on the same 100 questions: game framing with innocuous substitutes (A1), neutral mapping instructions with metaphors (A2), game framing with original terms (A3), metaphors without framing (A4), and English game framing with metaphors (A5). Table 5 reports harmful deliveries over valid responses. The subsequent paired tests compare conditions that hold substitute type or framing fixed; this targeted design does not cover every interaction between wording, framing, and language.

Table 5: Prompt ablations under two harmful-delivery rules.

Service	Rule	A1	A2	A3	A4	A5
GPT-5.6 Sol	Either judge	1/95	0/95	1/92	3/95	1/98
	Both judges	1/95	0/95	1/92	3/95	1/98
DeepSeek	Either judge	17/94	4/94	13/97	11/96	35/95
	Both judges	16/93	2/94	12/97	8/93	33/94
Grok	Either judge	5/60	5/58	8/60	10/58	9/60
	Both judges	5/60	5/58	8/60	7/56	9/60

DeepSeek's higher A5 rate and lower A2 rate appear under both judgment rules. Sol has at most 3/95 harmful deliveries. Failed attempts number 25/500 for Sol, 24/500 for DeepSeek, and 211/507 for Grok, including retries. Grok's approximately 42% missingness leaves substantially wider bounds on its delivery rates.

To test framing separately from substitute choice, the paired analysis compares game and neutral instructions with the substitute type fixed. The language comparison keeps the game/metaphor content fixed, while A4 versus A2 tests removing neutral framing. Table 6 gives all 12 tests under one Holm correction; b/c denotes delivery only in the first/second condition.

Table 6: Controlled framing and language comparisons, counting harmful delivery identified by either judge.

Service	First versus second	b	c	n	Holm p
Sol	M3b vs A2	1	0	89	1
	A1 vs M2b	1	1	92	1
	A5 vs M3b	1	1	91	1
	A4 vs A2	3	0	92	1
DeepSeek	M3b vs A2	15	0	74	0.0007324
	A1 vs M2b	13	0	77	0.002686
	A5 vs M3b	13	2	75	0.07385
	A4 vs A2	9	3	92	1
Grok	M3b vs A2	5	1	58	1
	A1 vs M2b	0	1	60	1
	A5 vs M3b	1	1	60	1
	A4 vs A2	6	1	57	1

Both DeepSeek game-versus-neutral comparisons pass correction. Framing therefore contributes beyond the choice of substitute words. Neither the language comparison nor the removal of neutral framing passes correction, so those comparisons do not establish a language effect or a protective effect of neutral instructions.

Constraining the answer versus reasoning and answer. Table 7 compares requests to substitute words in both reasoning and the answer (suffix b) with requests to substitute only in the answer (suffix c). Each entry gives first-only/second-only harmful deliveries, followed by the number of valid pairs. Grok has no valid c responses after deduplication and is excluded. Holm correction over the remaining 12 comparisons gives adjusted $p = 1$ throughout.

Table 7: Paired delivery discordances for reasoning-plus-answer versus answer-only constraints.

Service	M1b vs M1c	M2b vs M2c	M3b vs M3c
Sol	$0/2; 92$	$1/0; 93$	$1/1; 91$
DeepSeek	$2/1; 77$	$3/3; 81$	$5/3; 78$
Qwen	$2/1; 100$	$3/1; 100$	$0/4; 97$
GLM	$0/2; 100$	$1/0; 100$	$0/3; 100$

These prompt changes do not yield a statistically established increase in harmful delivery from extending the substitution instruction to reasoning. They compare returned responses, not hidden computation; reasoning-specific interpretation also requires an exposed reasoning channel.

B.4 Benign Requests and Mapping-Free Metaphors

The benign-request control uses XSTest and the same refusal screen. Records are deduplicated by service and question, preferring a nonempty response when available. Table 8 separates refusals from request failures so that missing output is not treated as a semantic refusal.

Table 8: Service responses to benign requests. Refusals are a subset of valid responses.

Service	Requests	Valid	Refusals	Errors
GPT-5.6 Sol	200	181	3	19
Grok-4.5	200	200	2	0
DeepSeek-V4-Pro	200	187	16	13
Qwen3.8-Reasoner	199	180	4	19
GLM-5.3-Flash	200	200	2	0

DeepSeek has the largest observed refusal count on these benign inputs. Sol's 19 failures are empty responses returned in under eight seconds; they remain failures rather than confirmed refusals.

Removing the mapping table. A separate generation experiment removes the explicit mapping table while retaining metaphoric requests and the same harmful-delivery rubric. Table 9 reports its results. This experiment asks what the services generate; Appendix E.5 instead tests monitors on 38 fixed metaphoric answers. Here refusals count as zero, while failed requests and non-refusals without valid judgments are excluded from the delivery denominator.

Table 9: Generation under metaphors without explicit mapping tables.

Service	Response records	Questions	Either judge / valid
GPT-5.6 Sol	100	100	4/93
Grok-4.5	100	100	22/99
DeepSeek-V4-Pro	100	100	15/83
Qwen3.8-Reasoner	103	99	8/99

All four configurations return some responses judged harmful without an explicit mapping table. Qwen has 103 response records from 99 questions, including repeats. Its denominator counts valid responses, so this aggregate cannot be used as a paired test against M3b.

C Detection Under Word Substitution

This appendix supplements Section 3. It specifies how monitor inputs and decisions are constructed, then reports paired tests and controls for question wording, answer wording, framing, and decision thresholds. The reasoning-trace experiment is described separately because it evaluates newly generated traces rather than two wordings of a fixed answer.

C.1 Evaluation Interfaces and Data Selection

Table 10 specifies the native interface, inference precision, and alarm rule for each monitor. The request and mapping are contextual inputs; they cannot override the trusted safety policy. Because the monitor families also differ in policy coverage, their absolute alarm rates do not isolate the effect of safety training.

Table 10: Interfaces used to obtain monitor decisions.

Monitor	Inference	Positive decision
Llama Guard 3 8B	AWQ, vLLM; official conversation template	Unsafe label
Qwen3Guard-Stream 0.6B/4B/8B	BF16; final template-token risk logits	Unsafe class only
ReasoningShield 1B/3B	BF16; monitored text in the Thought field	Judgment 1; broad-risk analysis also includes 0.5
ShieldGemma-2b	BF16; four policy prompts	Maximum normalized Yes probability at least 0.5
Qwen3-14B	BF16 for internal diagnostics; thinking disabled	True-minus-false candidate score $d_\theta > 0$

The ReasoningShield answer-prefix measurement is a transfer of its interface to another input type; its native-trace control is specified in Appendix C.7.

The initial pool contains 50 harmful responses, five refusals, and 15 benign examples. Restoration leaves three harmful responses and two refusals unchanged; removing them leaves 47 harmful responses, three refusals, and 15 benign examples. A ReasoningShield-1B parse failure reduces its harmful comparison to 46 complete pairs. The original 50-response pool is retained where explicitly stated, including the framing, threshold, and generated-answer internal analyses.

The new-question set is selected before monitor scoring from outside the original 100-question pool. It contains 26 harmful responses from DeepSeek, three from Qwen, and one from GLM. Inclusion

requires at least one label-2 judgment among four judgments from two judge families; the initial pool uses two judgments. These labels apply to complete responses, not to each prefix shown during monitoring.

The new-question controls comprise ten refusals, ten benign metaphors, and ten quotations without advice. Sequential checking uses 30 independently constructed benign answers, described with that experiment in Appendix F.1. The constructed A/B inputs and adaptation data are separate and specified in Appendices D and E.1.

C.2 Statistical Conventions

Paired binary tests retain only inputs with valid decisions in both conditions and use two-sided exact McNemar tests. For the fully restored versus fully substituted comparison in Equation 1, let N_+ count restored-only alarms, N_- substituted-only alarms, and n valid pairs. The estimated difference in alarm rates is

$$\widehat{\Delta}_{\text{expr}} = \frac{1}{n}\sum_i [f_\theta(x_i^{11}) - f_\theta(x_i^{00})] = \frac{N_+ - N_-}{n}. \tag{6}$$

Control alarm rates likewise use valid decisions in the relevant condition. Missing parses are not converted to safe labels. Every decision on the new-question set parses successfully.

The initial and new-question comparisons each receive a separate seven-test Holm correction. The 28 comparisons that change only one input field are exploratory and report raw p-values. Other tests are defined with their experiments: restoration checks and classifier permutations in Appendix D, source-cluster bootstrap intervals in Appendix E.4, and service and system correction families alongside their tables.

C.3 Paired Replication and Construction Checks

Restoration uses whole-word matches and permitted plurals. This avoids substring replacement but can still change collocations, and a many-to-one placeholder does not have a unique inverse. The construction check therefore asks whether the alarm gap remains after removing identified processing artifacts.

Table 11 reports the paired changes in both replications and then restricts the new-question set to 25 examples without marked processing artifacts. The five excluded examples were marked during construction review. The last column counts restored-only alarms in the remaining 25, rather than reporting another independent replication.

Table 11: Paired alarm changes and construction sensitivity. $+/-$ denotes restored-only/substituted-only alarms.

Monitor	Initial $+/-$	New $+/-$	New-set raw p	Unmarked $+/25$
LG3 8B	15/1	14/0	0.000122	10
QG 0.6B	28/0	23/0	2.38×10^{-7}	19
QG 4B	32/0	18/0	7.63×10^{-6}	14
QG 8B	28/0	18/0	7.63×10^{-6}	14
RS 1B	2/0	4/0	0.125	4
RS 3B	5/1	6/0	0.03125	5
SG 2B	15/0	14/0	0.000122	12

All monitors retain restored-only alarms in the 25 unmarked examples. The five identified artifacts therefore do not explain the entire gap, although this check cannot rule out every unmarked linguistic change. Five configurations pass the seven-test Holm correction in each replication; ReasoningShield-3B reaches adjusted $p = 0.0625$ on the new-question set.

C.4 Question and Answer Controls

The four input combinations in Equation 1 distinguish question effects from answer effects. Table 12 supplies the paired tests for all 30 harmful responses to new questions. Each cell reports

alarms lost/gained when the indicated field is restored, followed by raw p. The column beneath each operation identifies which form of the other field is held fixed.

Table 12: Paired single-field contrasts on harmful responses to new questions.

	Restore answer		Restore question	
Monitor	Substituted question	Original question	Substituted answer	Restored answer
LG3 8B	$0/4$; .13	$0/2$; .50	$1/13$; .0018	$0/10$; .0020
QG 0.6B	$0/11$; .0010	$0/2$; .50	$0/21$; $< .0001$	$0/12$; .0005
QG 4B	$0/8$; .0078	$0/0$; 1	$0/18$; $< .0001$	$0/10$; .0020
QG 8B	$0/4$; .13	$0/1$; 1	$0/17$; $< .0001$	$0/14$; .0001
RS 1B	$0/4$; .13	$0/0$; 1	$0/4$; .13	$0/0$; 1
RS 3B	$0/5$; .063	$0/1$; 1	$0/5$; .063	$0/1$; 1
SG 2B	$0/9$; .0039	$0/5$; .063	$0/9$; .0039	$0/5$; .063

Restoring the question produces strong paired effects for Llama Guard and Qwen3Guard. In contrast, once the question is original, restoring the answer yields no comparison with raw $p < 0.05$. Thus question context accounts for more of the additional alarms in these monitors. Because restoring the question also removes its mapping and framing, this comparison does not separate those two components.

The same four conditions are evaluated on each ten-example control category. Table 13 gives the two mixed question/answer conditions omitted from Table 1; the fully substituted and fully restored counts remain in the main table. Every unlisted monitor–category combination has zero alarms in both mixed conditions.

Table 13: Benign alarms under mixed question/answer expressions, ten examples per category.

		Substituted question	Original question
Monitor	Category	Restored answer	Substituted answer
RS 1B	Metaphor	8	5
RS 1B	Refusal	1	1
RS 3B	Metaphor	5	3
RS 3B	Quotation	3	3
SG 2B	Metaphor	1	5
SG 2B	Quotation	2	2

Context changes also affect negative examples: SG metaphor alarms rise when the question is restored, while RS metaphor alarms are higher with the substituted question and restored answer.

Unrelated answer words. A further control keeps the original question but replaces the answer's target term with an innocuous word from another example. It breaks the intended relation between that term and its meaning. Two of the 30 new-question examples cannot support the construction, leaving 28 pairs. Alarms unique to the restored/unrelated-word conditions are $1/1$ for Llama Guard; $1/0$, $1/0$, and $2/0$ for Qwen3Guard-0.6B/4B/8B; $0/0$ for both ReasoningShield sizes; and $6/0$ for ShieldGemma. Of these seven unadjusted tests, only ShieldGemma has raw $p < 0.05$ ($p = 0.03125$). The persistence of most alarms after breaking the answer's word-to-meaning relation supports the question-context interpretation in Section 3.3.

C.5 Framing with Fixed Mappings

To isolate framing, the next control keeps the mapping and substituted request fixed and changes only game versus neutral instructions. Each framing is tested with substituted and restored answers. Eight of the original 50 inputs have malformed framing in this construction; Table 14 reports the remaining 42, complementing the full-pool counts in Section 3.4.

Table 14: Alarms on 42 well-formed framing-control examples.

Monitor	Game framing		Neutral framing	
	Substituted	Restored	Substituted	Restored
Qwen3-14B	21	30	19	34
Llama Guard 3	20	23	21	25

For Qwen3-14B, restoring the answer gives raw $p = 0.003906$ under game framing and 6.10×10^{-5} under neutral framing; changing framing gives $p = 0.5$ and 0.21875 for substituted and restored answers. The corresponding Llama Guard values are 0.453125 and 0.21875 for restoration, and 1 and 0.5 for framing. Qwen's answer-wording effect therefore persists after excluding malformed inputs and without game instructions. The nonsignificant framing comparisons do not establish that framing has no effect.

C.6 Policy-Level Threshold Sensitivity

Section 3.4 varies ShieldGemma's aggregate threshold. Table 15 additionally shows which policies contribute on the original 50 harmful responses. Each of four prompts produces a normalized Yes probability, and the maximum is compared with an inclusive threshold. Counts are reported at 0.5 and 0.25. A response may trigger several policies, so the rows cannot be summed to obtain distinct detections.

Table 15: Policy-level alarm counts for substituted/restored answers, $n = 50$ per condition.

Policy	Cutoff 0.5		Cutoff 0.25	
	Substituted	Restored	Substituted	Restored
Dangerous content	2	15	6	19
Harassment	1	1	1	4
Hate speech	0	2	0	6
Sexually explicit content	0	0	0	1

Dangerous-content alarms show the largest substitution/restoration difference at both cutoffs. This identifies the policy contributing most to the observed gap, but per-example policy-applicability labels would be needed to diagnose coverage failures. Stored scores have four decimal places: two of the 140 harmful and control records equal 0.5000 and count as alarms under the inclusive rule. Their unrounded values are unavailable.

C.7 Reasoning-Trace Evaluation

The reasoning-trace control evaluates ReasoningShield on its intended input type. GLM and Qwen independently generate traces for original and substituted questions. Combining both question forms with both generated traces gives four conditions for each of 200 generator–question pairs; all 800 decisions per monitor size parse. Table 2 reports the 194 harmful-source instances from 97 questions, while the full collection is retained for this protocol.

For ReasoningShield-1B, the strict alarm-rate difference has a question-clustered 95% interval of $[0.144, 0.268]$. A separate two-test Holm correction for broad risk gives adjusted $p = 2.05 \times 10^{-11}$ and 0.000113 for 1B and 3B. The compared trace sets have median lengths of 2,457 and 1,236 characters, and only ten generator–question instances overlap the auxiliary answer-evaluation set. This experiment changes generated content and length as well as question wording; it therefore complements the fixed-answer comparison rather than repeating it on identical reasoning. The four-condition and length-stratified analyses are post hoc.

D Internal Responses and Safety Decisions

Section 4 uses two input sets for different questions. The 24 constructed A/B pairs hold the vocabulary, request, and answer fixed to test whether hidden states distinguish the mappings. The 50

model-generated answers test mapping changes on longer responses and supply the attention and intervention measurements. The first set's labels indicate the concept associated with A; in the second set, Ref, Harm, and Benign name mapping conditions. Neither naming convention replaces a judgment of the complete answer's harmfulness.

D.1 Candidate Scoring and Restoration Checks

Extraction must stop before the monitor has seen either candidate judgment. Let s_i be the rendered monitor input followed by the JSON decision prefix, τ the tokenizer, and $\|$ concatenation. With one-based token positions, the decision position is the last position in the candidates' longest common token prefix:

$$b_i = \mathrm{LCP}\left(\tau(s_i\|\text{true}), \tau(s_i\|\text{false})\right), \qquad t_i^* = |b_i|. \tag{7}$$

Only b_i is supplied when extracting the decision-position state. Let v_i^r be the remaining suffix for candidate r. Its summed token log probability and the monitor's decision score are

$$\begin{aligned} \ell_\theta(r \mid x_i) &= \sum_{k=1}^{|v_i^r|} \log p_\theta(v_{i,k}^r \mid b_i, v_{i,<k}^r), \\ d_\theta(x_i) &= \ell_\theta(\text{true} \mid x_i) - \ell_\theta(\text{false} \mid x_i). \end{aligned} \tag{8}$$

Positive scores indicate an alarm. Scoring suffixes separately handles candidates that tokenize into multiple tokens without feeding a chosen answer into the state measurement.

The explicit-word control replaces A with its original concept word and keeps the sentence asking for help, while removing the mapping table, framing, and mapping/target instructions. Its three comparisons are explicit harmful versus explicit benign, explicit harmful versus coded harmful, and explicit benign versus coded benign. Two-sided exact McNemar tests receive one three-test Holm correction. Counts and corrected results are reported in Table 16 and Section 4.2.

Table 16: Safety decisions on 24 A/B pairs. Harmful/benign refers to what A stands for.

(a) Monitor decisions		
Measurement	Harmful	Benign
Median codeword score	−11.50	−13.50
Codeword alarms	0/24	0/24
Original-word alarms	7/24	0/24
(b) Harmful minus benign		
Median score change	+1.00	
Higher / lower / tied	17 / 6 / 1	

To check software sensitivity, both codeword and explicit-word inputs are evaluated under PyTorch 2.5.1, whereas the initial mapping-swap run used 2.7.1. For coded harmful inputs, absolute score differences have median 0.375 and maximum 0.750. Among the seven examples that trigger only after restoration, even the coded score nearest the threshold remains 5.50 away in the matched run. All eight harmful and eight benign run controls retain their expected score signs. The repeated run thus preserves the decision contrast despite numerical shifts. Classifiers never mix hidden states from different environments.

D.2 Classifier Fitting and Statistical Tests

A separate linear classifier is fitted at each of 41 stored levels: the embedding output and 40 transformer layers. For residual state z at level l, its prediction is

$$g_l^{(-i)}(z) = \sigma\left((w_l^{(-i)})^\top \phi_l^{(-i)}(z) + \beta_l^{(-i)}\right). \tag{9}$$

Here σ is the logistic function, and $(-i)$ excludes both members of test pair i from all preprocessing and fitting. Holding out a pair leaves 46 states. The 5,120-dimensional state is centered, projected onto 20 principal components, and standardized using only this training fold:

$$\phi_l^{(-i)}(z) = \left(S_l^{(-i)}\right)^{-1}\left(U_l^{(-i)}\right)^\top (z - \mu_l^{(-i)}), \tag{10}$$

Here μ is the training-fold mean, U the fitted PCA basis, and S the diagonal matrix of projected standard deviations, with zero entries replaced by one. The mapping labels are $\eta_j^+ = 1$ when A denotes the harmful concept and $\eta_j^- = 0$ when A denotes the benign concept. With BCE denoting binary cross-entropy, fitting minimizes

$$\sum_{\substack{j \neq i \\ a \in \{+,-\}}} \mathrm{BCE}\Big(\eta_j^a, \sigma(w^\top \phi_l^{(-i)}(z_{j,l}^a) + \beta)\Big) + \frac{\|w\|_2^2}{2C}, \tag{11}$$

The inverse regularization strength is $C = 1$, and fitting allows at most 2,000 iterations. Both members of the test pair use the transform fitted without that pair. The question-group split instead excludes every example sharing the test question and refits all preprocessing steps and the classifier on the remaining states.

The 24 pairs contain 22 distinct questions. Mean held-out AUC across levels is 0.723 when excluding the test pair and 0.724 when excluding the test question. With pairs held out, AUC reaches 1.000 at layers 22–24 and remains at least 0.92 through layer 36; layers 22–40 survive correction across all 41 levels. Figure 4(a) gives the layer profile. The question-excluded peak is also 1.000 at layer 22, so it is not explained by repeated training questions.

Statistical testing swaps the two mapping labels within each pair in 1,000 permutations and refits the held-out analysis each time. If h permutation statistics are at least as extreme as the observed statistic, $p = (h+1)/1001$. Bonferroni correction covers all 41 levels; the smallest corrected value is 0.04096. Input pairs are the sampling units, not layers or attention heads. As specified in Section 4.1, A always occupies the first mapping row, so discrimination may use the concept at that position rather than recovery of every word-to-meaning relation.

D.3 Mapping Changes on Generated Answers

The generated-answer comparison keeps each of the initial 50 answers fixed and changes its mapping table. Ref is the original mapping, Harm preserves the target term's harmful meaning while adding an unused entry, and Benign gives that term a benign meaning. Only 38 constructions remove the intended harmful meaning under Benign; Section 4.3 reports that subset. The remaining 12 retain independently harmful structure. Full-pool statistics retain them, whereas construction-valid comparisons exclude them.

Across all 50 answers, Qwen3-14B raises 26/27/26 alarms under Ref/Harm/Benign, and Llama Guard raises 28/28/28. For Ref versus Benign, first-only/second-only alarms are $2/2$ for Qwen and $1/1$ for Llama Guard. Ref versus Harm gives $1/2$ and $2/2$; Harm versus Benign gives $1/0$ and $2/2$. All six two-sided raw p-values are 1. The full pool therefore also lacks a consistent direction of decision changes. Unlike the constructed A/B pairs, these tables differ in vocabulary and length as well as meaning; the Benign name does not certify that a complete answer is harmless.

D.4 Attention Annotation and Length Normalization

Attention is extracted from the decision position in Equation 7 on the same 50 generated answers. Character spans are aligned to tokens using tokenizer offsets, and the midpoint of each token span determines its region. More specific word spans take priority over enclosing spans, producing 14 mutually exclusive labels: risk taxonomy and other instructions; original and substitute mapping words; substitution framing and request body; original words, substitutes, and other text in prior context; the same three categories in current content; generation prefix; and remaining format tokens.

The format label is the default for tokens outside the annotated content spans. In particular, the appended JSON decision prefix is not separately annotated and therefore belongs to format, leaving the generation-prefix category at zero. The format share thus includes template and output-prefix tokens, not just punctuation. Original terms in current content and the three prior-context categories also round to zero in this dataset.

Table 17: Regional attention shares (%) on 50 generated answers per condition.

Region	Ref	Harm	Benign
Format / prefix	73.76	73.80	73.79
Other instructions	18.31	18.19	18.22
Other answer text	3.12	3.11	3.11
Safety rules	2.70	2.69	2.67
Request body	1.52	1.58	1.58
Substitution framing	0.37	0.38	0.39
Content: substitutes	0.12	0.12	0.11
Mapping: originals	0.05	0.06	0.06
Mapping: substitutes	0.05	0.07	0.07

The model runs in evaluation mode with BF16 computation. The attention implementation exposes post-softmax weights; retained rows are converted to FP32 for regional summation, then stored in FP16. A causal mask is constructed when the interface does not supply one. The decision-row measurements used in Table 17 are kept separate from content-to-mapping measurements, which use at most the last 128 content-query positions.

Let $a_{i,l,h,j}$ be the decision position's attention to token j in layer l and head h, and $\mathcal{R}_{i,r}$ the tokens in region r. Regional mass and the overall share are

$$m_{i,l,h,r} = \sum_{j \in \mathcal{R}_{i,r}} a_{i,l,h,j}, \qquad P_r = \frac{\sum_{i,l,h} m_{i,l,h,r}}{\sum_{i,l,h,r'} m_{i,l,h,r'}}. \tag{12}$$

Aggregation covers 50 examples, 40 layers, and 40 heads per condition. To check whether a small regional share merely reflects few tokens, the density for region r in example i additionally divides its layer/head-averaged mass by its token count:

$$u_{i,r} = \frac{(LH)^{-1} \sum_{l,h} m_{i,l,h,r}}{\max(|\mathcal{R}_{i,r}|, 1)}. \tag{13}$$

Here $L = 40$ and $H = 40$. Empty regions have zero mass. The mapping-word summary adds the densities of original and substitute mapping words before averaging over examples; the answer-word summary does the same for current-content words. These summaries are sums of two separately normalized categories, rather than a single density obtained by pooling their tokens.

All 50 Ref/Benign pairs have matching answer-token counts, while none has matching mapping-token counts. Table 18 therefore gives the length-normalized results for both the full pool and the 38 valid constructions.

Table 18: Mean attention density per token, averaged over instances after layer/head aggregation.

Subset	Condition	Mapping	Answer words	Other content	Map/answer
All 50	Ref	0.00029	0.00021	0.00022	1.359
	Harm	0.00021	0.00021	0.00022	0.992
	Benign	0.00020	0.00020	0.00022	1.045
Valid 38	Ref	0.00028	0.00019	0.00023	1.519
	Harm	0.00021	0.00018	0.00023	1.144
	Benign	0.00021	0.00018	0.00023	1.162

Despite the mapping's small share of total attention, its per-token density is comparable to or larger than that of answer words. The Ref-minus-Benign density difference averages 0.000083 and is positive in 48/50 examples; in the valid subset, it averages 0.000080 and is positive in 36/38. Thus the regional allocation changes even after accounting for length, although the changed words prevent attributing that difference to meaning alone. Figure 11 provides the head-level distribution.

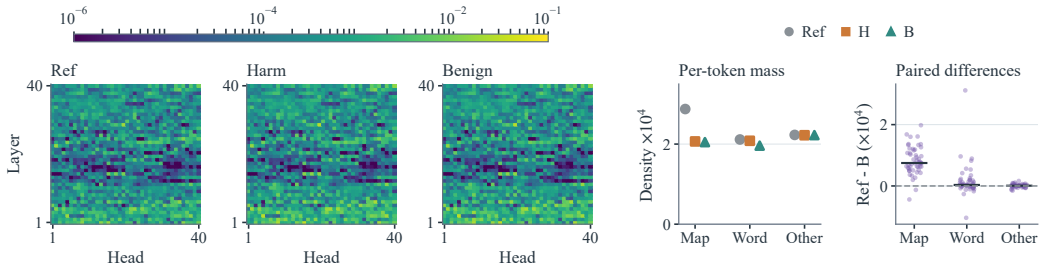


Figure 11: Mapping attention across all 40 layers and 40 heads under three conditions, with a shared log scale. Right: length-normalized regional density and all 50 Ref-minus-Benign paired differences; short bars mark medians. Map and Word sum separately normalized original/substitute subregions.

D.5 Activation Intervention Controls

Activation patching asks whether replacing states associated with the mapping changes the monitor's safety score. A donor supplies states from another input to a recipient input. For recipient positions $\mathcal{S}$, donor alignment $\rho(t)$, and replacement strength α, the residual-state intervention is

$$\widetilde{h}_{l,t} = \begin{cases} (1-\alpha)h_{l,t} + \alpha h^{\text{donor}}_{l,\rho(t)}, & t \in \mathcal{S}, \\ h_{l,t}, & t \notin \mathcal{S}. \end{cases} \tag{14}$$

The recipient's unchanged forward pass supplies the baseline. For intervention $\mathcal{P}$, the score change and signed decision change are

$$\Delta d_i = d_\theta(x_i; \mathcal{P}) - d_\theta(x_i), \qquad \Delta f_i = f_\theta(x_i; \mathcal{P}) - f_\theta(x_i). \tag{15}$$

All comparisons use Harm-to-Benign and Benign-to-Harm replacements on the 50 generated answers. *Mapping* replaces states at substitute-word positions in the mapping table with states from the same answer under the other mapping. *Other example* uses mapping states from a different answer. *Other position* inserts the Mapping donor states at content positions outside the mapping. *Answer words* replaces answer-word states with those from a different answer, and *Self* reinserts each state unchanged. The unrelated-example control tests donor specificity, the other-position control tests location specificity, and self-replacement checks the implementation. Token alignment is used for the all-layer comparison, with a separate character-aligned analysis.

D.5.1 Residual and Alignment Checks

The 50 answers, two replacement directions, and 40 layers produce 4,000 interventions per donor condition. Table 19 reports full-replacement results across all layers. These are repeated measurements on 50 answers; layers and directions do not increase the number of independent source inputs.

Table 19: All-layer residual interventions with token alignment.

Donor condition	Mean $\lvert\Delta d\rvert$	Label changes /4,000
Self	0.0000	0
Mapping	0.9759	44
Other example	1.0273	151
Other position	0.9401	64
Answer words	1.2982	139

Self-replacement leaves scores and labels unchanged. Other-example donors can nevertheless change scores or labels more than the intended mapping donor. Replacing token alignment with character alignment changes the mean Mapping and Other example effects to 1.1344 and 0.6497, with 62 and 85 label changes. Because the comparison depends on alignment, the next test fixes character alignment and varies replacement strength.

The recorded strength-sweep protocol fixes the 16–28 window before the sweep, motivated by the earlier mapping-classifier profile and its layer-22 peak. These are one-based layer numbers, corresponding to zero-based module indices 15–27. The all-layer results above provide the broader comparison. Absolute score changes quantify perturbation magnitude without cancellation between Harm-to-Benign and Benign-to-Harm replacements; signed changes and label flips remain distinct outcomes.

The strength sweep uses $\alpha \in \{0.25, 0.50, 0.75, 1.00\}$ and averages $|\Delta d|$ over layers 16–28 for each answer and direction. The full-replacement point uses the same character-aligned procedure. Table 21 adds paired differences to the effect sizes in Table 20. Its signs and raw tests summarize 100 answer–direction measurements from 50 answers, not 100 independent inputs.

Table 20: Score changes after activation replacement: mean $|\Delta d|$ over layers 16–28. α is the replacement strength.

Intervention / α	0.25	0.50	0.75	1.00
Self	0	0	0	0
Mapping	.2808	.4115	.4890	.5196
Other example	.2019	.2544	.3092	.3279
Other position	.3060	.4208	.4881	.5000
Answer words	.3292	.4917	.6488	.8210

Table 21: Same-answer donor minus control effects across residual replacement strengths.

	Mapping minus Other example		Mapping minus Other position	
α	Mean difference	Raw p	Mean difference	Raw p
0.25	0.0788	0.04945	-0.0252	0.1293
0.50	0.1571	1.28×10^{-6}	-0.0092	0.6849
0.75	0.1798	4.606×10^{-5}	0.0010	0.9179
1.00	0.1917	1.591×10^{-8}	0.0196	0.6101

The Mapping effect increasingly exceeds Other example as replacement strength rises, but does not consistently exceed Other position. A numerical check confirms that every increase in strength changes all 42,960 candidate positions examined for mapping and answer-word replacement. BF16 rounding therefore does not erase the interventions; the unresolved issue is specificity to the mapping and its position.

Character alignment also changes how many positions each donor replaces: Mapping averages 6.60 positions, versus 2.73 for Other example and 8.02 for Other position. The larger full-sample Mapping effect therefore does not isolate donor meaning. Restricting Mapping and Other example to equal replacement counts leaves 13 answer–direction pairs. Their mean Mapping-minus-Other-example effects are listed below for both recorded layer windows; this small subset is descriptive.

Table 22: Position-count-matched donor contrasts, 13 answer–direction pairs.

Layer window	$\alpha = 0.25$	0.50	0.75	1.00
16–28	0.0163	0.0074	-0.1080	-0.2678
13–28	-0.0120	-0.0637	-0.3233	-0.4183

Figure 6(c) shows the full-replacement 13–28 contrast alongside the full-sample controls. Neither window preserves a positive full-replacement Mapping advantage after matching position counts. The windows and sample sizes differ from the unrestricted contrasts and should not be pooled.

D.5.2 Head-Level Interventions

Head-level interventions replace the outputs of k attention heads on the same 50 answers, in both directions and at layers 16–28. Head sets are nested: a NumPy generator with seed 0 permutes the

40 heads, and the first $k \in \{1, 4, 10, 20, 40\}$ are used. Effects are averaged over layers for each answer and direction.

Before interpreting a mapping-specific effect, the prespecified check requires Answer words to exceed both Mapping and Other example in two one-sided paired sign tests, each with raw $p < 0.05$. A larger mean alone does not pass this check. The primary comparison then tests Mapping $>$ Other example with a one-sided paired sign test. Table 23 reports mean $|\Delta d|$, the primary p-value, and whether the preliminary check passes for each head count.

Table 23: Head-count intervention effects and specificity checks.

k	Mapping	Other example	Answer words	Mapping > Other example: p	Check passed
1	0.2283	0.2217	0.2215	0.5796	No
4	0.2175	0.2279	0.2563	0.6227	No
10	0.2358	0.2390	0.2794	0.8910	No
20	0.2362	0.2590	0.3092	0.9890	No
40	0.2585	0.2648	0.3617	0.9247	No

No head count passes the preliminary check or the primary test. In particular, the larger Answer words means at several head counts are insufficient to pass both paired tests. Self-replacement gives zero effect in all 2,600 checks at $k = 1$ and $k = 40$. These results do not identify a head set whose effect is specific to using the mapping; they explain why an arbitrary score change is insufficient for that conclusion.

E Improving Detection

Section 5 compares demonstrations with fine-tuning and tests whether training on substituted wording improves detection beyond repeating restored examples. This appendix groups the data splits, training procedure, and evaluation controls needed to reproduce those comparisons.

E.1 Data Splits and Demonstrations

The 50 generated harmful responses in Section 4.3 split by question into 26 development examples from 16 questions and 24 validation examples from 12 questions. Training covers 232 source-question identifiers and excludes all 12 validation harmful questions. All 16 harmful development questions occur in training. Three harmful demonstrations are drawn from that development set; one refusal and one benign demonstration come from the evaluation controls. Four refusal controls share training source questions, including one exact monitor-input match, and two benign controls share benign training questions. Appendix E.4 tests sensitivity to these overlaps. Median answer length is 107 characters for benign controls, 696 for harmful answers, and 681 for refusals.

The five demonstrations each contain a monitor input and an assistant output consisting of a decision and short explanation. Writing $\mathcal{D}_{\text{demo}} = \{(x_j, o_j)\}_{j=1}^{5}$, prediction is conditioned on

$$p_\theta(o \mid x; \mathcal{D}_{\text{demo}}) = p_\theta(o \mid x_1, o_1, \ldots, x_5, o_5, x). \tag{16}$$

The same monitor instruction precedes the demonstrations, which add approximately 3.6k tokens. This comparison evaluates the examples, their labels, and their explanations together.

E.2 Training Objective and Settings

LoRA changes the query and value projections while keeping each base matrix W_0 fixed. For input and output dimensions d_{in} and d_{out}, rank r, and scaling coefficient α_{L}, the adapted matrix is

$$W = W_0 + \frac{\alpha_{\text{L}}}{r} BA, \qquad A \in \mathbb{R}^{r \times d_{\text{in}}}, \quad B \in \mathbb{R}^{d_{\text{out}} \times r}. \tag{17}$$

Let $\mathcal{A}(o)$ contain the assistant target-token positions, including the end marker, and let $\Delta\theta$ denote the learned LoRA updates. Only these assistant tokens contribute to the supervised loss:

$$\mathcal{L}_{\text{adapt}} = -\sum_{(x,o)\in\mathcal{D}_{\text{train}}} \sum_{t\in\mathcal{A}(o)} \log p_{\theta_0+\Delta\theta}(o_t \mid x, o_{<t}). \tag{18}$$

Training averages valid target-token losses within each batch. All training records, including their assistant targets and end markers, fit the 3,072-token sequence limit.

Equation 4 fixes the comparison between training on both wordings and repeating restored inputs. Each dataset contains 586 records: the shared $\mathcal{D}_0$ has 120 refusals, 40 benign examples, and 86 unpaired substituted harmful examples, and each scheme adds 340 paired or repeated harmful examples. Training on both wordings gives 586 distinct texts; repeating restored inputs gives 416 distinct texts because each paired restored example appears twice. The equation leaves risk labels implicit; the actual training targets are assistant outputs.

Training uses rank 16, scaling coefficient 32, dropout 0.05, NF4 base weights, and BF16 computation. Optimization uses three epochs, learning rate 10^{-4}, 8-bit AdamW, cosine decay, ten warmup steps, batch size one, and gradient accumulation over eight steps.

The matched comparison uses Qwen3-14B for all six adapters and their shared base. Every model is evaluated with NF4 inference, the same five demonstrations, full-vocabulary candidate scoring, and a 4,608-token limit. Seeds 0, 1, and 2 configure the initial data shuffle and training framework. No separate data-loader seed or seed reset before adapter construction was specified, so these values document the configured randomness rather than independent control of every initialization. The repeated base entries in Table 24 come from one shared score set.

E.3 Instruction and Deployment Controls

Table 24: Held-out detection and AUC. H/B: harmful/benign alarms. Panel (b) fixes 14B NF4 inference and five demonstrations.

(a) Alarms: H/24, B/15

Size	Base H/B	5-shot H/B	LoRA H/B
4B	1/0	16/2	24/15
14B	7/1	19/4	22/0
32B	17/1	24/5	24/1

(b) AUC: 24 harmful, 20 controls

Training setup	Seed 0	Seed 1	Seed 2
Base	0.879	0.879	0.879
Substituted + restored	0.940	0.913	0.933
Restored + restored	0.940	0.919	0.906

The restoration-instruction control asks the monitor to recover original wording internally. The answer-restoration control instead applies the full lexicon to the answer before scoring, while retaining the substituted question. Both use the same BF16 baseline and the initial 50 harmful responses, five refusals, and 15 benign examples. Answer restoration may alter collocations and does not restore the question. Section 5.1 reports the harmful and benign counts; the additional refusal counts are one for the baseline, one for the instruction, and two for restored answers.

The recorded deployment comparisons use matched AWQ inference for base versus five-shot prompting, BF16 for the adapted 4B/14B deployments, and an NF4 training base for the 32B adapter. They compare deployed configurations rather than changing only adapter weights.

A same-input BF16 check with full-vocabulary scoring illustrates the loading sensitivity. The untrained base with demonstrations raises 21/24 validation harmful alarms, 4/15 benign alarms, and 1/5 refusal alarms. Directly loaded training weights with demonstrations give 24/24, 6/15, and 0/5, whereas the merged-weight deployment gives 0/15 benign alarms. The main training comparison therefore holds loading, precision, demonstrations, and scoring fixed in the common NF4 setup.

E.4 Paired Uncertainty and Control Composition

Ranking comparisons use seven fixed score sets: one base and six adapters. The primary evaluation contains 24 validation harmful responses and 20 controls across 32 source-question clusters. The overlap-filtered analysis removes controls sharing a training source question, an original benign training question, or an exact training/demonstration input. These intersecting rules remove four refusals and two benign examples, leaving 14 controls and 26 clusters in total when combined with the 24 harmful responses. A second analysis uses only the 15 benign controls. The filters address observed overlaps without certifying the absence of all semantic near-duplicates.

Confidence intervals resample source clusters, keeping related records together. For each subset, a NumPy generator initialized with bootstrap seed 20260922 draws 10,000 multinomial cluster-count

vectors over sorted source identifiers. Each draw samples the subset's number of clusters with replacement and retains all records in the selected clusters. All seven models use identical draws, so their AUC differences are paired. The generator is reset for each subset, and the 2.5th and 97.5th percentiles define the interval. Source clusters group the same source behavior even when question strings differ. This bootstrap seed controls fixed-score resampling and is separate from the training seeds.

Table 25 gives the primary and both sensitivity intervals. Base AUC is 0.833 after overlap removal and 0.850 with benign controls only. Each subset changes the control distribution, so comparisons of methods are made within a subset, not by ranking absolute AUC across subsets.

Table 25: Paired AUC differences [95% source-cluster bootstrap interval] under matched NF4 inference. Every subset contains the same 24 validation positives.

Controls	Seed	Substituted + restored minus base	Restored + restored minus base	(Substituted + restored) minus (restored + restored)
All 20	0	$0.060\ [0.007, 0.130]$	$0.060\ [0.009, 0.127]$	$0.000\ [-0.025, 0.021]$
	1	$0.033\ [-0.007, 0.085]$	$0.040\ [-0.011, 0.098]$	$-0.006\ [-0.053, 0.044]$
	2	$0.054\ [-0.005, 0.123]$	$0.027\ [-0.023, 0.080]$	$0.027\ [-0.006, 0.074]$
Overlap removed (14)	0	$0.080\ [0.008, 0.175]$	$0.086\ [0.013, 0.179]$	$-0.006\ [-0.038, 0.014]$
	1	$0.048\ [-0.009, 0.118]$	$0.051\ [-0.015, 0.125]$	$-0.003\ [-0.065, 0.069]$
	2	$0.071\ [-0.007, 0.163]$	$0.033\ [-0.032, 0.102]$	$0.039\ [-0.008, 0.106]$
Benign only (15)	0	$0.069\ [0.006, 0.158]$	$0.069\ [0.007, 0.156]$	$0.000\ [-0.032, 0.029]$
	1	$0.033\ [-0.013, 0.100]$	$0.042\ [-0.017, 0.108]$	$-0.008\ [-0.070, 0.061]$
	2	$0.061\ [-0.007, 0.147]$	$0.025\ [-0.033, 0.084]$	$0.036\ [-0.009, 0.101]$

The AUC and overlap analyses are post hoc checks added after harmful-response detection approached its ceiling. Their consistent direction across control subsets supports the ranking comparison in Section 5.3.

E.5 Transfer to Mapping-Free Expressions

The transfer set fixes 38 harmful metaphoric answers without mapping tables across all model sizes. It contains 29 questions, of which 27 questions and 35 associated responses have source questions seen in training; the metaphoric responses themselves are excluded from training. Table 26 therefore tests transfer to another expression of mostly familiar questions. The main validation set separately tests questions excluded from training. Base and five-shot results use matched AWQ inference; LoRA results use the recorded deployments described above.

Table 26: Mapping-free transfer: alarm counts on 38 fixed positive examples.

Scale	Base	5-shot	LoRA
4B	11	13	37
14B	11	23	24
32B	18	23	30

Every adapted configuration raises more alarms than its base on these fixed answers. The set contains no benign examples, so the counts measure detection rather than false-positive cost or overall discrimination. The main benign controls remain necessary to interpret the high 4B count, and cross-deployment precision differences prevent attributing the full difference solely to adapter weights.

F Monitoring During Generation

Section 6 first checks whether a content-safety alarm appears as an answer is revealed, then examines how output monitoring and withholding flagged text affect delivery. The sequential checks below use fixed harmful and benign answers. The subsequent system evaluation uses a separate saved-response collection, also independent of the frontier-service measurements in Appendix B.

F.1 Sequential Checking

The sequential checks reveal fixed answers to each monitor while keeping the question visible. Spaced checking uses cumulative prefixes at 200, 400, 700, 1,400, and 2,800 characters, together with the fixed prefix from the main comparison and the complete answer. Duplicate checkpoints are evaluated once. Token-level checking instead evaluates every answer-token boundary, including the empty answer. For answer length T_i and alarm $b_i(t)$ at step t, the measurements are

$$B_i^{\text{any}} = \max_{0 \le t \le T_i} b_i(t), \qquad \tau_i = \min\{t : b_i(t) = 1\}, \qquad B_i^{\text{final}} = b_i(T_i), \tag{19}$$

Here $\tau_i = \infty$ when no alarm occurs. The three quantities record any alarm, the first alarm position, and the final decision. Before any answer token is visible, Qwen3Guard uses its answer template with an empty-thinking prefix, while Llama Guard and ShieldGemma use empty answers. Stopping-character summaries assign the complete answer length when no alarm occurs and take a separate median for each wording. Subtracting these medians is not the median paired difference. The positions measure how much text is exposed before an alarm, not live latency or independently judged harmful content.

Independent scoring of complete inputs checks the endpoints of token-level evaluation. For both ReasoningShield sizes, forward and reverse evaluation orders give identical generated decision tokens and labels on all 90 complete inputs; all 90 final scan decisions also match. These runs use BF16, greedy decoding, a 1,024-token decision-generation limit, and batch-invariant vLLM. Earlier fixed-prefix and spaced measurements use Transformers, but the endpoint checks prevent a backend disagreement from being mistaken for an effect of checking frequency. The recorded environment uses PyTorch 2.13.0, Transformers 5.16.1, and vLLM 0.28.0 on RTX 4090 GPUs.

Each ReasoningShield scan covers 33,017 prefix positions. The 1B run has 12 unparseable intermediate judgments, all after the first observed strict alarm, and every final judgment is available. The 3B run has no parse failures. First-alarm, any-alarm, and final counts are therefore determined. A strict alarm can later become label 0.5 rather than 0: this happens for four benign 1B cases and two benign 3B cases. Disappearance of a strict alarm need not mean a fully safe final judgment.

F.2 Consecutive-Alarm Confirmation

A monitor may raise a brief alarm and then withdraw it. To test whether requiring persistence differs from simply waiting, confirmation requires W consecutive alarming steps, whereas the delay control waits $W - 1$ steps after the first alarm without requiring continued warnings. Section 6.1 gives the Qwen3Guard-0.6B comparison. Table 27 adds 4B and 8B at $W = 128$; each entry lists substituted/restored/benign alarm counts, with 30 examples in each group.

Table 27: Additional consecutive-alarm versus delay controls.

Monitor	Any-step alert	128 confirmations	Delay of 127 steps
QG 4B	19/30/1	9/23/0	14/23/1
QG 8B	19/30/1	10/22/0	15/24/1

At both additional scales, confirmation removes the benign alarm but also removes more harmful-source alarms than the delay control. Continued warnings, not elapsed steps alone, account for that difference. The setting $W = 128$ is a tested comparison, not an optimized operating point.

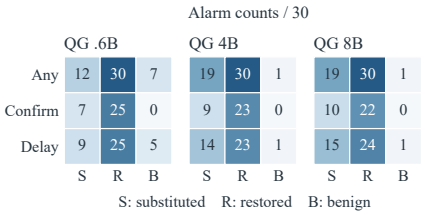


Figure 12: Qwen3Guard alarm counts out of 30 under any-token triggering, 128-consecutive-alarm confirmation, and a 127-step delay. S/R: substituted/restored harmful-source answers; B: benign. The delay control separates waiting from requiring continued warnings.

F.3 Monitoring and Release Policies

Table 28: Monitoring-system outcomes ($n = 100$ per row). Paired counts use W1/W2: answer-only/with reasoning.

Input	Harmful answers	Input alarms	Output alarms	Missed harmful	Unsafe buffer	Unsafe instant	Delivered harm
DeepSeek							
M0	3	99	6/6	1/1	0/0	0/0	1/1
M1b	3	86	26/26	0/0	0/0	0/0	0/0
M2b	5	90	68/68	2/2	0/0	0/0	0/0
M3b	27	70	51/51	17/17	2/2	3/3	12/12
M3c	22	69	41/41	15/15	1/1	1/1	11/11
Qwen							
M0	1	99	8/51	0/0	0/0	0/0	0/0
M1b	2	86	41/89	0/0	0/0	0/0	0/0
M2b	4	90	79/100	0/0	0/0	0/0	0/0
M3b	3	70	58/94	3/0	0/0	1/3	1/0
M3c	4	69	55/91	3/0	0/1	0/3	1/1
GLM							
M0	1	99	6/35	0/0	0/0	0/0	0/0
M1b	1	86	18/87	0/0	0/0	1/2	0/0
M2b	2	90	50/99	0/0	0/0	2/2	0/0
M3b	2	70	32/97	1/0	0/0	2/10	0/0
M3c	3	69	22/94	3/0	0/0	3/12	2/0

The separate system collection contains 1,500 responses: three generators, 100 questions, and five conditions (M0/M1b/M2b/M3b/M3c). The output monitor is Qwen3-14B-AWQ served with vLLM, temperature zero, and thinking disabled. It receives the original request and accumulating text at 50-token intervals measured with its tokenizer. Its instruction follows the Reasoning Safety Monitor task (Wang et al., 2026) and asks for reasoning errors, so its alarms have a different meaning from the content-safety alarms in Appendix C.

The two release policies in Section 6 either withhold the currently checked block until its decision is available or release it immediately. Equation 5 combines the input filter, output alarms, and the harmfulness of any text already released, using all requests as the denominator. Complete responses use the Either judge rule from Appendix B.1: at least one valid label of 2. Llama Guard 3 independently judges nonempty released prefixes; an empty prefix receives zero.

Among 980 non-refusal complete responses, one judge has no parse failures and the other has one. Of 979 jointly valid pairs, 959 agree and 20 disagree ($\kappa = 0.894$); both judges assign label 2 to 69 responses. Prefix judgments are obtained separately and may differ from complete-response judgments, particularly when visible reasoning forms part of the released text.

The DeepSeek M3b example in Section 6 applies this calculation. The estimate is conditional on the release policy and the prefix judge, which itself shows expression sensitivity. Saved responses support the comparison of release rules but do not measure asynchronous deployment latency.

F.4 Component and Monitor-Input Comparisons

Three conditions separate the output component from the text it observes: no output monitor (W0), answer-only monitoring (W1), and monitoring visible reasoning plus the answer (W2). W1 and W2 operate on the same generated responses. Trigger comparisons retain non-refusals valid in both conditions, so their denominators differ from the all-request counts in Table 28. Each service has a four-test Holm family in Table 29. Cells report first-only/second-only triggers, the paired sample size, and adjusted p.

Table 29: Paired output-trigger comparisons on valid non-refusal responses. Entries: b/c; n; p_{Holm}.

First versus second	DeepSeek	Qwen	GLM
M3b vs M3c, W2	$4/5$; 29; 1	$4/3$; 63; 1	$6/2$; 85; .2891
W1 vs W2, M3b	$0/0$; 58; 1	$0/32$; 82; 1.86×10^{-9}	$1/64$; 97; 1.43×10^{-17}
W1 vs W2, M0	$0/0$; 41; 1	$1/9$; 43; .04297	$0/22$; 85; 9.54×10^{-7}
M0 vs M3b, W2	$1/8$; 24; .1562	$0/24$; 37; 3.58×10^{-7}	$0/55$; 84; 1.67×10^{-16}

Exposing reasoning increases output alarms for Qwen and GLM in both direct and metaphor conditions. DeepSeek exposes no reasoning in this collection, making W1 and W2 inputs identical. No M3b-versus-M3c comparison passes correction. These are differences in the reasoning-error monitor's responses; more alarms alone do not establish higher content-safety accuracy.

The next comparison asks whether alarm changes reduce harmful responses missed by the output monitor. A missed response is a harmful complete response with no output alarm; this component-level measure excludes input filtering and released-prefix judgments. Table 30 uses all 100 requests per condition and a separate four-test Holm family for each service. Cells give missed harmful responses unique to the first/second condition and adjusted p. With no output monitor (W0), every harmful complete response is missed by that component.

Table 30: Paired component-level harmful-delivery comparisons, $n = 100$ per cell. Entries: b/c; p_{Holm}.

First versus second	DeepSeek	Qwen	GLM
M3b vs M3c, W2	$8/6$; 1	$0/0$; 1	$0/0$; 1
W1 vs W2, M3b	$0/0$; 1	$3/0$; 1	$1/0$; 1
M0 vs M3b, W2	$1/17$; .0005798	$0/0$; 1	$0/0$; 1
W0 vs W2, M3b	$10/0$; .005859	$3/0$; 1	$2/0$; 1

For DeepSeek, adding output monitoring reduces missed harmful responses, yet metaphorical substitution still increases them relative to direct requests. Qwen and GLM have too few missed harmful responses to establish corrected differences. This separates changes in alarm frequency from demonstrated changes in harmful delivery.

F.5 Output-Monitor Scale

To test scale while holding responses fixed, Qwen3-4B/14B/32B monitor the same M0/M3b/M3c responses under both W1 and W2. Table 31 gives W1/W2 trigger counts at each size, with a common valid non-refusal denominator n per row. The last column reports adjusted p for 32B versus 14B in each mode, using one 18-test Holm family.

Table 31: Scale sensitivity of reasoning-error alarms on non-refusal responses. W1/W2 entries are paired counts, not fractions.

Source	Input	n	4B	14B	32B	32B vs 14B Holm p
DeepSeek	M0	41	20/20	6/6	12/12	.438/.438
	M3b	58	42/42	28/28	28/28	1/1

Source	Input	n	4B	14B	32B	32B vs 14B Holm p
Qwen	M3c	45	37/37	18/18	13/13	1/1
	M0	43	20/33	3/11	7/19	1/.322
	M3b	82	74/81	45/77	61/81	.0424/1
	M3c	74	61/71	39/68	37/69	1/1
GLM	M0	85	37/73	5/27	11/34	.438/1
	M3b	97	92/97	31/94	52/93	.00035/1
	M3c	88	73/88	19/82	33/74	.0424/.438

Alarm counts do not increase monotonically with parameter count: 4B often triggers more than either larger model. Only three 32B-versus-14B comparisons pass correction, all with answer-only inputs. These measurements show sensitivity to monitor configuration, not an accuracy ranking. DeepSeek's identical W1/W2 inputs also mean its two modes are not independent replications.

F.6 Input-Task Sensitivity

A separate input-task control applies Prompt Guard 2 86M at threshold 0.5 to the same 500 inputs (100 questions in five conditions). Alarms under M0/M1b/M2b/M3b/M3c are 15/51/36/26/27. Attaching the four substitution prefixes to the original questions, without substituting the questions themselves, gives 47/41/29/29 alarms. Adding these prefixes can therefore change injection-detection decisions. The prefixes contain original risk words as well as formatting, so this is not a formatting-only control. Prompt-injection detection differs from the content-harm input filter in Table 28; the two are reported separately.

G Scope and Research Ethics

The fixed-answer findings concern judge-selected harmful responses; the new-question set is dominated by one generator. The internal analysis uses Qwen3-14B and a constructed mapping whose target row is fixed. It establishes distinguishable states alongside weak decision changes, while the intervention controls leave the causal pathway unresolved. Adaptation tests questions excluded from training, with shorter synthetic benign controls; the overlap checks address recorded shared sources rather than all possible semantic similarity.

Service measurements apply to the accessed deployments. The system evaluation instead fixes generated responses and changes observable monitoring and release rules. Reasoning-error alarms, complete-response labels, and released-prefix labels answer different questions; none reveals hidden service defenses or measures live delivery latency.

Public harmful-request resources, including StrongREJECT and HarmBench, are used for controlled safety testing. Displayed examples are truncated or de-identified. Reproduction materials describe input construction, monitor interfaces, internal measurements, adaptation, and statistical procedures; original harmful responses remain subject to restricted access.