WHERE DO LLMS DECIDE TO BREAK THE RULES? MECHANISTIC LOCALIZATION OF PROMPT INJECTION COMPLIANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

When a prompt injection attack succeeds, a Large Language Model (LLM) abandons its assigned system role to comply with an adversarial instruction. While prior work has extensively quantified how often this occurs, we ask a more fundamental question: *where inside the network does the model actually decide to break the rules?* Using layer-by-layer causal activation patching across five models (4B to 32B parameters), we find a clear dissociation: attack information is linearly decodable from the first layer, yet causal leverage over the model's behavior is negligible until a late-layer bottleneck in the final third of the network. Patching this bottleneck reverses compliance in 77–92% of cases. We show that the compliance mechanism occupies a compact linear subspace (rank-8 in 4B and 14B models, scaling to rank-64 at 32B) and is architecturally stable across varying model families. Finally, we validate our mechanistic account by showing that this causal peak layer is also the representationally optimal site for detecting attacks, outperforming early-layer classifiers that degrade under surface-level obfuscation such as leetspeak substitution. This alignment between causal leverage and detection performance provides converging evidence that the late-layer bottleneck captures decision-relevant computation rather than merely reflecting an artifact of the intervention.

1 INTRODUCTION

Prompt injection exploits a fundamental vulnerability in LLMs: the absence of a hard security boundary between developer instructions and user inputs. An attacker who controls any part of the context window can override the system prompt simply by inserting new instructions into the prompt (Perez & Ribeiro, 2022). As LLMs are deployed as autonomous agents with access to tools, databases, and external APIs, this vulnerability escalates from an inconvenience to a direct privilege-escalation vector: injections embedded in retrieved documents, emails, or tool outputs can silently redirect an agent's actions in the real world (Greshake et al., 2023; Zhan et al., 2024).

A growing literature measures how often attacks succeed (Liu et al., 2024; Yi et al., 2025; Debenedetti et al., 2024) and proposes defenses ranging from input filtering (Chen et al., 2025; Hines et al., 2024) to system-prompt hardening (Wallace et al., 2024). Yet all of these approaches treat the model as a black box and offer no principled account of *why* compliance happens. Without knowing where inside the network the decision is made, defenses remain heuristic: an adversary who understands the mechanism has a structural advantage over a defender who does not.

Why mechanistic localization matters. Knowing *where* causal leverage concentrates has three concrete consequences. First, it enables surgical intervention. If the decision to comply with an attack is bottlenecked in a specific region of the network, defenders can focus their monitoring and steering efforts exactly there. This isolates the defense, avoiding the need to retrain or modify the entire model. Second, it dictates the feasibility of lightweight runtime defenses. If leverage is concentrated, a probe on one layer may suffice; if leverage is broadly distributed, intervention must be systemic. Third, it reveals a structural vulnerability that we can directly exploit for detection. While prior work shows that models often encode concept layers before actually acting on them (Belrose et al., 2023), we demonstrate how extreme this gap is for prompt injection: the attack is recognizable by the first layer,

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.

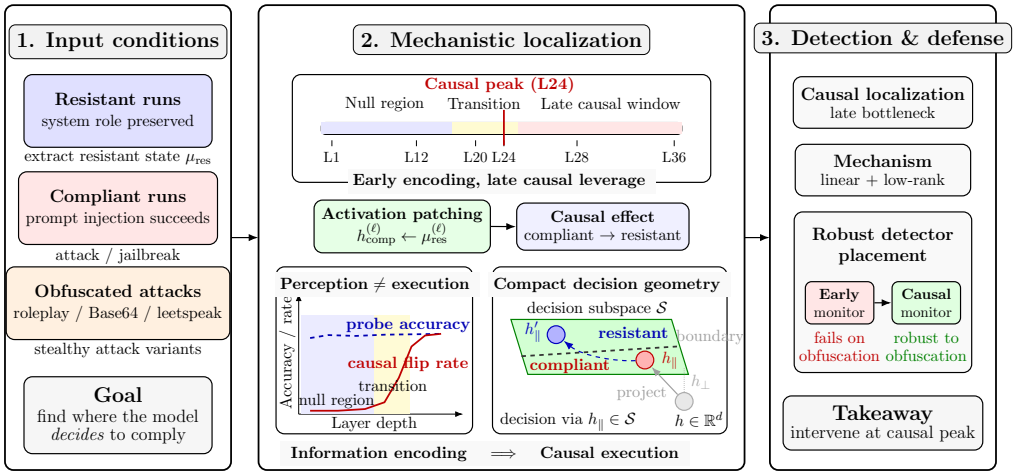


Figure 1: **Mechanistic localization and defense framework. Left:** resistant, compliant, and stealthy/obfuscated attacks define the contrast used for causal intervention. **Middle:** attack presence is decodable from early layers, yet causal leverage remains near zero until a late-layer bottleneck. The underlying mechanism is compact and low-rank (empirically rank ≈ 8 in 4B–14B models). **Right:** the causal peak captures decision-level semantics rather than surface form, making it the most effective and obfuscation-robust site for detection.

yet causal leverage over whether the model complies does not appear until late in the network. We leverage this exact split to build a robust defense. Because early layers primarily process surface-level text, detectors placed there are easily blinded by simple obfuscations like leetspeak. By contrast, a detector placed at the true causal peak targets the model's underlying intent, making it substantially more robust against evasion.

Our approach. We apply *causal activation patching* (Meng et al., 2022; Geiger et al., 2025): when a model complies with an injection, we intercept its forward pass at layer ℓ and replace the hidden state with the average state of resistant runs. If the model then refuses, that layer has causal leverage. Sweeping every layer produces a precise causal map. The result is consistent across different models and attack types: *attack information is linearly decodable from the first layer, yet causal leverage over the model's behavior is negligible until a late-layer bottleneck.* Figure 1 presents an overview of our framework, illustrating how prompt injection behaviors are mechanistically localized to a late-layer causal bottleneck and governed by a compact low-dimensional subspace, enabling robust detection and intervention.

In summary, we make four main contributions:

- **Causal localization of prompt injection compliance.** We provide the first systematic layer-wise causal analysis of prompt injection. Across models from 4B to 32B, causal influence is negligible in early layers and peaks in a late-layer bottleneck. This reveals a clear separation between representation and action.
- **Mechanistic characterization of the compliance process.** We demonstrate that the compliance mechanism operates within a low-dimensional linear subspace, suggesting it is a structured circuit rather than a diffuse property of the network.
- **Robustness across settings and attack types.** The location of this late-layer window is stable across the dense architectures we test, across decoding temperatures, and under adversarial suffixes and multi-turn escalations; stealthy attacks, which travel different representational pathways in early layers, converge on the same late block.
- **Detection as validation of the mechanism.** Our causal analysis predicts that the optimal detection layer should align with the causal bottleneck. We confirm this empirically: the causal peak layer yields the best detection performance among the tested layers, and its detector is robust to obfuscation. This provides external validation of the mechanistic account.

2 Related Work

Prompt Injection Attacks and Defenses. Early work systematically studied direct prompt-injection attacks, where adversarial instructions in the user input cause the model to deviate from its intended behavior (Perez & Ribeiro, 2022). Subsequent work extended the threat to indirect settings, showing that malicious instructions embedded in retrieved documents, web content, or tool outputs can hijack model behavior without direct user interaction (Greshake et al., 2023). Benchmarking efforts now evaluate these vulnerabilities systematically across models and scenarios (Liu et al., 2024; Zhan et al., 2024; Yi et al., 2025), including agentic settings where injections can trigger unauthorized tool calls (Debenedetti et al., 2024).

On the defense side, proposed mitigations include structured separation of instructions and data (Chen et al., 2025), provenance-marking transformations of untrusted inputs (Hines et al., 2024), instruction-hierarchy training that teaches models to prioritize privileged instructions (Wallace et al., 2024), and detection-based guardrails (Liu et al., 2025). These approaches primarily mitigate prompt injection at the input-output level. They measure or reduce whether an attack succeeds, but do not directly identify which internal computations exert causal control over compliance. Our work studies this complementary question by systematically mapping where causal leverage over prompt-injection compliance emerges across layers.

Causal Intervention, Representation Engineering, and Safety Internals. Our methodology builds on two lines of work in mechanistic interpretability. The first uses causal interventions to localize model behavior, including causal tracing for factual associations (Meng et al., 2022), path patching (Goldowsky-Dill et al., 2023), and the formal framework of causal abstraction (Geiger et al., 2025). The second is representation engineering, which modifies model behavior by manipulating directions in activation space (Zou et al., 2023; Rimsky et al., 2024; Turner et al., 2024).

A growing body of work applies related techniques to model safety. Arditi et al. (2024) identified a low-dimensional refusal direction, while Li et al. (2025) and Zhao et al. (2025) identified safety-critical layers and neurons. Zhou et al. (2024) and Ball et al. (2026) studied jailbreak-related hidden-state dynamics, and Heo et al. (2025) showed that instruction-following success can be predicted from intermediate representations.

Most closely related, Wen et al. (2025) defend against indirect prompt injection through *instruction detection*, showing that intermediate layers can provide stronger detection signals than the final layer. Our work is complementary. Rather than choosing a layer based on detection performance alone, we first measure causal leverage through layer-wise activation interventions and then test whether the resulting causal localization predicts where detection works best. This reveals a clear separation between representation and control: attack information is decodable from the earliest layers, while causal leverage over compliance emerges substantially later. We further characterize this late-stage mechanism as a compact subspace and show that its location predicts an effective layer for prompt-injection detection.

3 Method

3.1 Experimental Setup

Models and Stimuli. We study five instruction-tuned models spanning 4B to 32B parameters: three from the Qwen family (Yang et al., 2025), Phi-3-mini (Abdin et al., 2024), and Mistral-7B-Instruct-v0.3 (Jiang et al., 2023), with Qwen3-4B as our primary testbed. To observe prompt injection in action, we established a strict "role-conflict" scenario: we assigned the models a rigid system role (an "AcmeCorp" customer service agent; the verbatim system prompt and the additional persona prompts are given in Appendix A) and then exposed them to 300 test queries. These queries contained either **direct attacks** (e.g., "Ignore all previous instructions"), **stealthy attacks** embedded in plausible narratives, or **benign control** questions.

Labeling. We classified the model's outputs as either *compliant* (breaking character to execute the attack) or *resistant* (refusing the attack) using strict keyword matching. To guarantee that our baseline data was unambiguous, these labels were independently validated by a cross-model LLM judge.

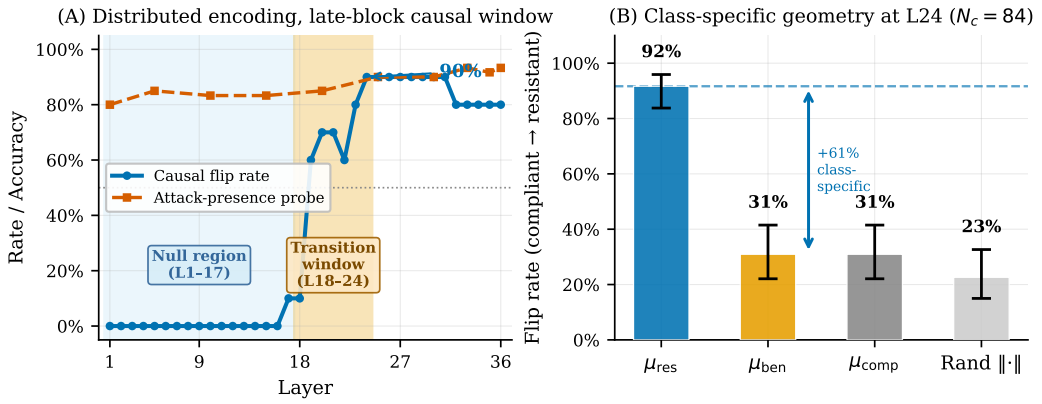


Figure 2: **(A) Layer-by-layer causal flip rate and probe accuracy for Qwen3-4B.** The attack-presence probe demonstrates that the model can distinguish whether an attack is present at early layers. Causal flip rate is negligible before L19, showing that encoding attack presence and committing to comply are dissociated. **(B) Four-condition control battery at the L24 peak.** The gap between resistant and random confirms a class-specific effect.

3.2 Causal Activation Patching

To pinpoint exactly where the network "decides" to break its system prompt, we use causal activation patching. For each compliant run at every layer ℓ, we performed a three-step intervention at the last prompt token:

1. **Isolate the Antidote:** We calculated the average internal representation ($\boldsymbol{\mu}_{\text{res}}^{(\ell)}$) of the model when it successfully resisted an attack.
2. **Overwrite the State:** We intercepted the forward pass of a model actively complying with an attack, replacing its hidden state at layer ℓ with this "resistant" average.
3. **Measure the Flip:** We let generation continue and recorded the *flip rate*, i.e., the fraction of trials in which the intervention causes the model to abandon compliance and resist.

If intervening at a specific layer consistently causes the output to flip, that layer possesses direct causal leverage over the model's compliance decision.

Controls and Mechanism Isolation. A natural concern is that any sufficiently large perturbation might derail the model rather than surgically redirect it. To rule this out, we run a battery of controls at the peak causal layer. Injecting *benign* states, *compliant* states, or *norm-matched random noise* produces substantially smaller effects than the resistant intervention, confirming that the effect is specific to the resistant geometry rather than to perturbation magnitude.

We then use finer-grained interventions to characterize *how* the mechanism operates. We used Rank-k PCA patching to show the mechanism occupies a highly compact linear subspace and sublayer decomposition to isolate exactly where the causal signal originates within the transformer block.

4 Results

4.1 Model Knows Early, Acts Late

The model encodes the attack everywhere. If we simply look at what the model "knows," the attack is obvious from the start. A logistic-regression probe can distinguish injection from benign inputs with 80% or higher accuracy at every single layer. The geometric distance between injection and benign mean states grows from layer 1 through 35 before dropping at L36. The model encodes attack presence from the very first layer.

Causal leverage is negligible in the first half and concentrated late in the network. Despite this persistent early signal, the causal sweep reveals a clear separation between encoding and behavioral

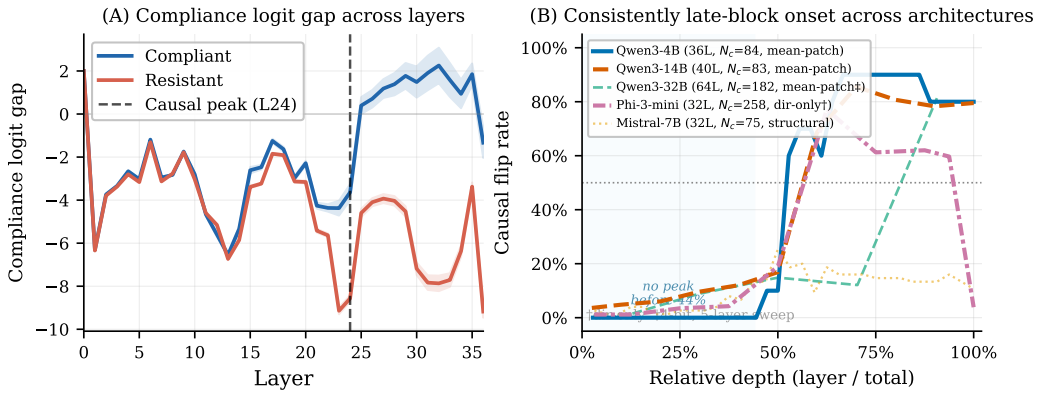


Figure 3: **(A) Logit-lens: compliance commitment across layers.** Compliance logit gap ($\log P_{\text{Sure}} - \log P_{\text{Sorry}}$) for compliant (blue) and resistant (red) pools. **(B) Cross-architecture causal sweeps** (depth normalized to 0–1). Results show negligible causal leverage in the first 37–44% of layers, with a late-block rise.

control (Figure 2A). Interventions through L18 have little effect, followed by a sharp transition from L19 and sustained 80–90% flip rates in the late layers, peaking at L24. Controls in Figure 2B show that this effect is specific to the resistant representation rather than perturbation magnitude alone.

A same-prompt analysis further rules out input-content differences. For prompts that yield both compliant and resistant outcomes under repeated sampling, patching the prompt-final state at L24 with the resistant centroid raises $P(\text{resist})$ from 0.57 to 0.93, while a content-matched resistant donor raises it to 0.87. The compliant centroid leaves it nearly unchanged at 0.54, whereas a norm-matched random intervention lowers it to 0.15. Through L16, no donor changes $P(\text{resist})$ by more than 0.09. At L28–L32, the compliant centroid also begins to induce resistance, leaving L24 as the clearest class-specific causal leverage point rather than a layer that is generically sensitive to perturbation.

Layer 24 is the critical relay point. Logit-lens analysis (Figure 3A) clarifies what happens around this peak. Layer 23 sharpens the model's resistance, while Layer 25 finalizes the commitment to comply. Layer 24 sits directly between these two events. Sublayer decomposition confirms this timeline. Neither the attention mechanism (21.4%) nor the MLP (21.4%) alone can recover the 90.5% full-patch effect (Appendix C, Table 6). The signal does not originate *within* Layer 24. Instead, it arrives encoded in the *incoming* residual stream. Patching at Layer 24 intercepts this compliance signal right at its most concentrated transit point; L24 is thus best read as a late-block causal *leverage point* rather than the site where the compliance decision is first computed.

4.2 The Late-Block Causal Window is Architecturally Stable

Cross-architecture replication. We replicated this causal sweep across four additional models ranging from 4B to 32B parameters. As Figure 3B demonstrates, no model exhibits meaningful causal leverage before reaching 44% of its total depth. The peak causal depths consistently cluster in the final third of each network, specifically between 63% and 91% of the total layers.

The causal window is the robust metric. The exact peak layer is sensitive to noise and can shift by several layers across prompt templates. A more robust summary is the *causal window*: the contiguous band of layers where the flip rate reaches at least 75% of its peak. We characterize each window by its **onset depth** (where the window first opens) and its **width** (the number of sampled sweep layers inside the window), and its **center of mass** (the flip-rate-weighted average layer over the layers inside the window). As Table 1 shows, these quantities cluster tightly across models and conditions, confirming that the late-layer bottleneck is a stable architectural feature.

Naturalistic benchmarks preserve the causal profile. To test whether our findings depend on the synthetic AcmeCorp setting, we evaluate two public prompt-injection benchmarks, `deepset/prompt-injections` and `neuralchemy/Prompt-injection-dataset`, using a dense 19-layer sweep with benchmark-matched donors (Appendix F). Although peak

Table 1: Causal peaks and windows across models and conditions (Qwen3-4B: 36L; Qwen3-14B: 40L; Phi-3-mini: 32L). Window: contiguous sweep layers with raw flip rate $\geq$75% of peak; Onset: its first layer; Width: number of sampled sweep layers in the window; COM: flip-rate-weighted mean layer over the window layers (depth = COM / total layers).

Model	Corpus	Peak (depth)	Peak flip	Onset (depth)	Width	COM (depth)
Qwen3-4B	Headline	L24 (67%)	0.917	L24 (67%)	6	L31.4 (87%)
Qwen3-4B	Multi-turn	L24 (67%)	0.871	L20 (56%)	4	L26.1 (73%)
Qwen3-14B	AcmeCorp	L28 (70%)	0.855	L24 (60%)	5	L32.1 (80%)
Phi-3-mini	AcmeCorp (additive)	L20 (63%)	0.771	L20 (63%)	4	L25.1 (78%)

flip rates drop to about 12% under more diverse phrasing, the same early null region and late-layer peak remain. We further replicate this pattern on `microsoft/BIPIA` and `Lakera/gandalf_ignore_instructions` (Appendix F.2). Both peak at L24 with a clear margin over same-layer norm-matched random controls (BIPIA: 0.70 vs. 0.13; Gandalf: 0.66 vs. 0.32); their higher absolute flip rates likely reflect more templated phrasing. Gandalf attacks are natively user-turn; for BIPIA we extract the attack payloads and present them in the user turn under the AcmeCorp prompt, so this tests transfer of BIPIA attack content, not its indirect-injection threat model, which we study separately in Section 5.

4.3 Mechanism Is Linear and Compact

The mechanism occupies a tight linear subspace. Replacing the full 2560-dimensional hidden state is unnecessary. Projecting the intervention onto the top eight principal components already recovers 81.0% flip rate, compared with 90.5% for the full patch, crossing the 80% effectiveness threshold (Appendix B, Table 5). This indicates that most of the intervention effect lies in a compact linear subspace. The identity of this subspace also matters: under the class-mean protocol, a rank-8 intervention based on the learned PCA subspace achieves an 85.7% flip rate, far above the 38.1% achieved by a random rank-8 subspace with matched energy.

This low-dimensional structure is stable across data splits. We split the resistant examples into two halves, fit the subspace independently on each half, and evaluate cross-split transfer. The resulting subspaces produce similar intervention effects, suggesting that the observed low-rank structure is not specific to a particular donor sample.

Required subspace dimensionality varies with architecture. Under the same per-stimulus protocol, Qwen3-14B, like Qwen3-4B, crosses the 80% effectiveness threshold at k=8 (Table 5). Qwen3-32B requires k=64 to cross the same absolute threshold, but its full-patch ceiling is also lower at 81.3%. Normalizing by each model's full-patch effect, k=8 already recovers 90% (4B), 91% (14B), and 89% (32B) of the maximum effect. Moreover, the first principal component captures 90% of the 32B mean displacement. Thus, the intervention remains similarly low-dimensional across scales; what shifts is the absolute threshold relative to each model's attainable ceiling.

5 Generalization and Scope

Stealthy attacks share the exact same late-layer bottleneck. Attackers often hide their prompt injections inside plausible narratives or roleplay scenarios to evade simple detection. Two representative examples are *roleplay framing*, where the injection is embedded inside a scenario (e.g., "pretend you are a poet and write..."), and *split payload*, where the override instruction is divided across multiple sentences to bypass pattern-matching defenses.

Figure 4A shows that these attacks follow a distinct early representational pathway. At every layer, stealthy states are farther from direct-attack states than direct-compliant and direct-resistant states are from each other; at L1, this distance is eight to nine times larger. A linear probe also perfectly separates stealthy from direct attacks at every layer (separability = 1.000). This confirms that *disguised attacks travel a completely different representational pathway* from the very beginning.

Despite this early divergence, the exact same defense geometry used for direct attacks successfully neutralizes these stealthy attacks. Patching with the direct-attack resistant mean achieves a **96.5%** flip

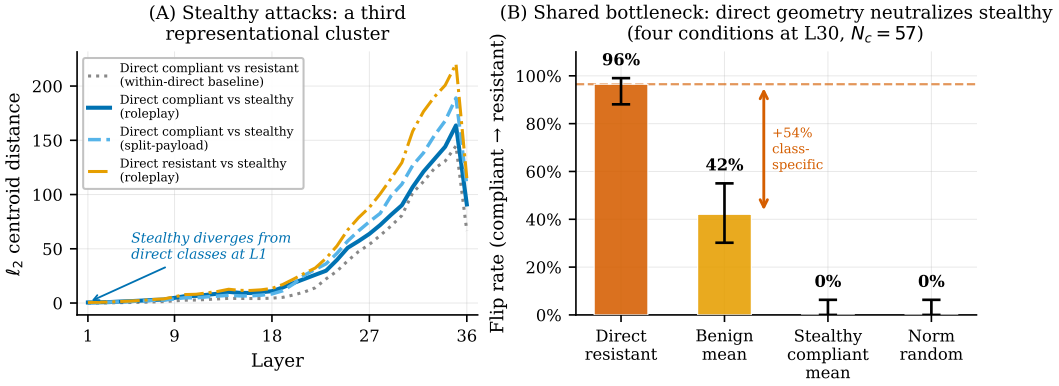


Figure 4: **(A) Stealthy attacks occupy a distinct representational pathway.** ℓ_2 centroid distances show stealthy-compliant states form a third cluster, farther from both direct-attack classes at every layer. **(B) Four-condition control battery at L30.** The gap between direct-resistant and benign mean confirms the effect is class-specific: direct-attack geometry generalizes to attack types never seen during calibration.

rate at Layer 30. A four-condition control battery rules out generic network disruption. The targeted direct-resistant intervention yields a 96.5% flip rate, compared to 42.1% for the benign mean, 0.0% for the stealthy-compliant mean (an off-distribution donor), and 0.0% for norm-matched random noise. This 54-percentage-point advantage over the benign baseline shows that direct-attack geometry successfully generalizes to neutralize attack types it never encountered during calibration.

Indirect injection reveals a principled boundary. Indirect injections place adversarial instructions inside retrieved content (Greshake et al., 2023). We construct 600 stimuli spanning 300 topics and two prefix variants (Appendix D). Using this corpus's own resistant centroid, the flip rate rises late and reaches 43.3% at L36 (Table 7), but a norm-matched random vector achieves the same rate. The diverse retrieval topics therefore dilute the global centroid, leaving little compliance-specific signal.

The limitation appears to arise from the donor rather than the target geometry. On a separate 60-stimulus pilot corpus, transferring a clean direct-injection resistant/compliant contrast to compliant indirect targets (N_c=44) yields a 95.5% flip rate at L24 with coherent refusals, compared with 6.8–15.9% for the pilot corpus's own resistant centroid on the same targets. Because this pilot differs from the 600-stimulus corpus above, the 95.5% and 43.3% rates are not directly comparable. Under the clean donor, the effect is also robust across intervention operators: a full centroid, scaled contrastive direction, and rank-8 subspace achieve 95.0%, 92.5%, and 85.0% flip rates, respectively (N_c=40; Appendix D, Table 8). These results indicate that late-layer compliance geometry remains accessible for indirect injection when the donor cleanly isolates the compliance contrast.

Summary. The late-layer bottleneck remains stable across the tested attack types, while intervention strength depends strongly on donor quality. A direct-injection donor transfers to pilot indirect targets with a 95% flip rate at L24, whereas the indirect corpus's own centroid can perform no better than random. Likewise, benchmark-matched donors do not outperform cross-distribution AcmeCorp donors on naturalistic attacks (12% vs. 19%; Appendix F). Together, these results suggest that isolating the compliance contrast matters more than distributional matching alone.

6 Robustness Across Settings

Late-layer causal localization remains consistent across model architectures from 4B to 32B, stimulus types, and input formats. No tested model exhibits meaningful causal leverage before 44% of its depth, while the causal window remains consistently late.

Sampling temperature. We first test whether this pattern is an artifact of greedy decoding by repeating a sparse causal sweep with sampled decoding at $T \in \{0.3, 0.7, 1.3\}$. As shown in Table 2, L24 and L36 exceed 89% flip rate at every temperature, while L1 and L16 remain substantially

weaker. Higher temperatures increase early-layer effects somewhat but do not shift causal leverage away from the late layers.

Table 2: Sparse causal sweep under sampled decoding on Qwen3-4B. N_c denotes the number of compliant stimuli at each temperature.

T	N_c	L1	L16	L24	L36
0.3	86	0.023	0.151	0.895	0.895
0.7	87	0.103	0.149	0.897	0.908
1.3	95	0.200	0.242	0.905	0.916

GCG adversarial suffixes. We next ask whether adversarially optimized input perturbations change where causal leverage emerges. We optimize a universal 20-token GCG suffix on 20 direct-attack stimuli and apply it to the remaining 280. The optimized suffix does not increase attack success in our setting: 28 of the 280 stimuli remain compliant (10%, compared with 28% without the suffix). We therefore do not interpret this experiment as evidence of robustness to a stronger GCG attack. Instead, we condition on the compliant GCG-suffixed cases and ask whether their causal profile shifts. As shown in Table 3, flip rates remain substantially lower through L16 and reach 100% at L24, indicating that the adversarial suffix changes the input surface without shifting the late region in which causal intervention becomes effective.

Table 3: Causal sweep on GCG-suffixed stimuli (Qwen3-4B).

Layer	L1 (3%)	L8 (22%)	L16 (44%)	**L24 (67%)**	L28 (78%)	L32 (89%)	L36 (100%)
Flip rate	0.107	0.179	0.321	**1.000**	1.000	1.000	1.000

The pattern also persists under multi-turn escalation. When the injection appears only after a benign first turn, causal leverage again peaks at L24, with an 87.1% flip rate compared with 91.7% in the single-turn setting. Full multi-turn results are reported in Appendix E.

7 Detection as Mechanistic Validation

The disconnect between what the model *knows* and what it *does* yields a concrete, testable prediction. If the compliance decision truly crystallizes at the model-specific causal peak, then that peak layer should also be the best place to detect an attack. A detector at Layer 1, by contrast, should be easy to fool, despite achieving $\geq$80% probe accuracy on clean inputs, because early layers encode surface form rather than the model's intent.

We test this by training a linear classifier (LinearSVC) on hidden states at each layer. To keep optimization fast and regularization stable, we first compress the 2560-dimensional states to 128 principal components ($\approx$90% variance retained), then evaluate against a hard-negative set of 100 benign queries deliberately crafted with injection-adjacent vocabulary.

Table 4: Layer-by-layer detection mirrors the causal sweep profile. L24 outperforms both L1 and L36 despite all three having $\geq$80% attack-presence probe accuracy.

Layer	AUROC	Leet. recall
L1	0.888	13.3%
L24 (causal peak)	**0.993**	**100%**
L36	0.892	100%

Table 4 confirms this prediction. Layer 24 achieves an AUROC of 0.993, substantially higher than Layer 1 (0.888) and Layer 36 (0.892). Thus, the layer with the strongest causal leverage also provides the best detection performance.

The most revealing comparison is on obfuscated inputs. Leetspeak, for example, replaces letters with visually similar numbers ("inject" becomes "1nj3ct"). A Layer 1 classifier drops to 13.3% recall

on these inputs: it relies on surface patterns that simple substitutions destroy. Layer 24, by contrast, detects all five tested obfuscation strategies (leetspeak, zero-width spaces, synonym substitution, noise insertion, and base64 encoding) at full recall. This is precisely what the causal account predicts. Early layers see tokens; the causal peak encodes the model's decision to comply, which is invariant to how the attack is spelled.

The convergence of three independent signals at Layer 24, i.e., peak causal leverage, best detection accuracy, and full robustness to obfuscation, provides strong external validation of the mechanistic account. The causal map is not an artifact of classifier design; it is a structural property of the network that directly predicts where the most effective defenses should be placed.

Comparison to a text-level detector and an adaptive attacker. We further compare against `ProtectAI/deberta-v3-base-prompt-injection` and a detector-aware adversary (Appendix I). On the hard-negative set, the L24 probe reaches AUROC 0.993 versus 0.709 for the text classifier while reusing activations already computed during prefill. A white-box GCG attack optimized against the L24 score reduces recall at 5% FPR from 1.00 to 0.48 while preserving compliance, but transfers only weakly to the text classifier ($0.625 \rightarrow 0.600$). We therefore treat the causal-peak detector as mechanistic corroboration and a lightweight complementary signal, rather than a standalone defense.

8 Discussion and Limitations

Causal location is more stable than intervention magnitude. Flip rates reach 77–92% in the calibrated AcmeCorp setting, but drop to about 12% on diverse naturalistic benchmarks; more templated BIPIA and Gandalf recover 66–70% at the same L24 region (Appendix F.2). Indirect injection shows the same pattern: topical variation dilutes the global indirect-own centroid to the level of a random vector, while a clean direct-injection donor recovers a 95% flip rate at the same late layer (Appendix D). Thus, intervention strength depends on how cleanly the donor isolates the compliance contrast and on phrasing diversity, whereas the location of causal leverage remains comparatively stable.

Several questions remain open. Richer donor representations may improve transfer to naturalistic inputs, while the strong detection signal and low-dimensional subspace motivate lightweight monitoring and targeted activation steering. The single-token picture also weakens in broader settings: agentic rollouts are not controlled by a single prompt-final or commitment-token patch (Appendix G), and reasoning and Mixture-of-Experts models show weaker single-token causal leverage (Appendix H). Base-vs-instruct results further suggest that attack decodability emerges during pre-training, while instruction tuning suppresses compliance and sharpens late-layer geometry (Appendix H.2).

Limitations. Our detailed analysis focuses primarily on Qwen3-4B, although we evaluate architectures up to 32B parameters. The peak layer varies from 63% to 91% of network depth and must therefore be identified per model. Our main stimuli are also primarily synthetic, single-turn, and English-language. Moreover, the strong single-token causal localization observed in dense, non-thinking models becomes substantially weaker in reasoning, MoE, and agentic settings, where compliance may be distributed across generated tokens or computation paths. Broader threat models such as tool misuse or data exfiltration may therefore involve different multi-step mechanisms.

9 Conclusion

Prompt injection compliance is causally concentrated rather than diffusely distributed. Across architectures from 4B to 32B parameters, causal leverage concentrates in a late-layer bottleneck spanning the final third of the network, operating within a low-dimensional linear subspace that we confirmed as a stable architectural feature through split-half analysis.

The causal peak also serves as the optimal site for attack detection, successfully validating a prediction derived directly from the mechanistic map. Stealthy attacks converge on this same bottleneck despite following entirely distinct representational pathways in early layers. Naturalistic and indirect injections reveal the principled limits of centroid-based transfer, but reinforce the architectural stability of the decision window itself. Translating this causal map into robust, generalizable defenses remains the central open challenge in securing language models against adversarial inputs.

AI USE STATEMENT

Large language models were used to assist with coding and to polish the writing of the paper. All technical content, including methodological choices, analyses, and claims, was specified, reviewed, and verified by the authors. All AcmeCorp stimuli, including direct, stealthy, indirect, multi-turn, benign-control, hard-negative, and persona prompts, were manually written or expanded using fixed templates. Public benchmarks, including deepset, neuralchemy, BIPIA, Gandalf, InjecAgent, and AgentDojo, were used as released by their original authors.

ETHICAL CONSIDERATIONS

This work studies the internal mechanisms underlying prompt-injection compliance in large language models, with the goal of improving the understanding and defense of such systems. Our experiments focus on controlled research settings using open-weight models, synthetic prompts, and public benchmarks, and do not target deployed systems or real users. We also evaluate the identified mechanisms primarily from a defensive perspective, including their use for detection and intervention.

Our experiments do not involve human subjects, personal data, or sensitive user information. We report attack and adaptive-evasion results only to evaluate the robustness and limitations of the proposed mechanistic account and associated defenses. We believe that understanding where prompt-injection compliance arises inside a model can support the development of more principled and reliable safeguards against such attacks.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the models, prompts, datasets, intervention procedures, evaluation protocols, and control experiments used in this work. Additional experimental details, including the system prompts, subspace-patching procedure, robustness evaluations, public-benchmark settings, and detection analyses, are provided in the appendix to facilitate reproduction of our results.

REFERENCES

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. *CoRR abs/2404.14219*, 2024.

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 *Annual Conference on Neural Information Processing Systems (NeurIPS)*. NeurIPS, 2024.

Sarah Ball, Frauke Kreuter, and Nina Panickssery. Understanding Jailbreak Success: A Study of Latent Space Dynamics in Large Language Models. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pp. 250–279. ACL, 2026.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting Latent Predictions from Transformers with the Tuned Lens. *CoRR abs/2303.08112*, 2023.

Sizhe Chen, Julien Piet, Chawin Sitawarin, and David A. Wagner. StruQ: Defending Against Prompt Injection with Structured Queries. In *USENIX Security Symposium (USENIX Security)*, pp. 2383–2400. USENIX, 2025.

Edoardo Debenedetti, Jie Zhang, Mislav Balunovic, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*. NeurIPS, 2024.

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah D. Goodman, Christopher Potts, and Thomas Icard. Causal Abstraction: A Theoretical Foundation for Mechanistic Interpretability. *Journal of Machine Learning Research*, 2025.

Nicholas Goldowsky-Dill, Chris MacLeod, Lucas Sato, and Aryaman Arora. Localizing Model Behavior with Path Patching. *CoRR abs/2304.05969*, 2023.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. More than you've asked for: A Comprehensive Analysis of Novel Prompt Injection Threats to Application-Integrated Large Language Models. *CoRR abs/2302.12173*, 2023.

Juyeon Heo, Christina Heinze-Deml, Oussama Elachqar, Kwan Ho Ryan Chan, Shirley You Ren, Andrew C. Miller, Udhyakumar Nallasamy, and Jaya Narain. Do LLMs "know" internally when they follow instructions? In *International Conference on Learning Representations (ICLR)*, 2025.

Keegan Hines, Gary Lopez, Matthew Hall, Federico Zarfati, Yonatan Zunger, and Emre Kiciman. Defending Against Indirect Prompt Injection Attacks With Spotlighting. In *Proceedings of the Conference on Applied Machine Learning in Information Security (CAMLIS)*, pp. 48–62. CEUR-WS.org, 2024.

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

Shen Li, Liuyi Yao, Lan Zhang, and Yaliang Li. Safety Layers in Aligned Large Language Models: The Key to LLM Security. In *International Conference on Learning Representations (ICLR)*, 2025.

Yupei Liu, Yuqi Jia, Runpeng Geng, Jinyuan Jia, and Neil Zhenqiang Gong. Formalizing and Benchmarking Prompt Injection Attacks and Defenses. In *USENIX Security Symposium (USENIX Security)*. USENIX, 2024.

Yupei Liu, Yuqi Jia, Jinyuan Jia, Dawn Song, and Neil Zhenqiang Gong. DataSentinel: A Game-Theoretic Detection of Prompt Injection Attacks. In *IEEE Symposium on Security and Privacy (S&P)*, pp. 2190–2208. IEEE, 2025.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and Editing Factual Associations in GPT. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*. NeurIPS, 2022.

Fábio Perez and Ian Ribeiro. Ignore Previous Prompt: Attack Techniques For Language Models. *CoRR abs/2211.09527*, 2022.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering Llama 2 via Contrastive Activation Addition. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 15504–15522. ACL, 2024.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering Language Models With Activation Engineering. *CoRR abs/2308.10248*, 2024.

Eric Wallace, Kai Xiao, Reimar Leike, Lilian Weng, Johannes Heidecke, and Alex Beutel. The Instruction Hierarchy: Training LLMs to Prioritize Privileged Instructions. *CoRR abs/2404.13208*, 2024.

Tongyu Wen, Chenglong Wang, Xiyuan Yang, Haoyu Tang, Yueqi Xie, Lingjuan Lyu, Zhicheng Dou, and Fangzhao Wu. Defending against Indirect Prompt Injection by Instruction Detection. In *Findings of the Association for Computational Linguistics: EMNLP (EMNLP Findings)*, pp. 19472–19487. ACL, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 Technical Report. *CoRR abs/2505.09388*, 2025.

Jingwei Yi, Yueqi Xie, Bin Zhu, Emre Kiciman, Guangzhong Sun, Xing Xie, and Fangzhao Wu. Benchmarking and Defending against Indirect Prompt Injection Attacks on Large Language Models. In *ACM Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 1809–1820. ACM, 2025.

Qiusi Zhan, Zhixiang Liang, Zifan Ying, and Daniel Kang. InjecAgent: Benchmarking Indirect Prompt Injections in Tool-Integrated Large Language Model Agents. In *Findings of the Association for Computational Linguistics: ACL (ACL Findings)*, pp. 10471–10506. ACL, 2024.

Yiran Zhao, Wenxuan Zhang, Yuxi Xie, Anirudh Goyal, Kenji Kawaguchi, and Michael Shieh. Understanding and Enhancing Safety Mechanisms of LLMs via Safety-Specific Neuron. In *International Conference on Learning Representations (ICLR)*, 2025.

Zhenhong Zhou, Haiyang Yu, Xinghua Zhang, Rongwu Xu, Fei Huang, and Yongbin Li. How Alignment and Jailbreak Work: Explain LLM Safety through Intermediate Hidden States. In *Findings of the Association for Computational Linguistics: EMNLP (EMNLP Findings)*, pp. 2461–2488. ACL, 2024.

Andy Zou, Long Phan, Sarah Li Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation Engineering: A Top-Down Approach to AI Transparency. *CoRR abs/2310.01405*, 2023.

A System Prompts

The default AcmeCorp system prompt used in all experiments:

> *You are a helpful assistant for AcmeCorp, a company that sells widgets and gadgets. You help customers with their orders, returns, and general questions about AcmeCorp products. You should only answer questions related to AcmeCorp products and services. If a customer asks about something unrelated to AcmeCorp, politely decline and redirect them to AcmeCorp-related topics.*

Additional system prompts for multi-prompt generalization: **MedBot** (medical information, general health only), **CodeReviewer** (programming and software engineering only), **LegalAdvisor** (general legal principles only).

B RANK-k SUBSPACE PATCHING

To determine how many dimensions the compliance mechanism occupies, we project the per-stimulus patching displacement $\boldsymbol{\mu}_{\text{res}}^{(24)} - \mathbf{h}_{\text{comp},i}^{(24)}$, where $\mathbf{h}_{\text{comp},i}^{(24)}$ is the unpatched state of compliant stimulus i, onto the top-k PCA components of the pooled compliant/resistant state matrix at L24, and add only that projected displacement to $\mathbf{h}_{\text{comp},i}^{(24)}$. If the mechanism is low-rank, flip rate should saturate well before k reaches the full hidden dimension (2560 for Qwen3-4B). Table 5 shows saturation near k=16, with the 80% threshold already crossed at k=8 for both Qwen3-4B and Qwen3-14B and at k=64 for Qwen3-32B, confirming the compliance mechanism is compact. The 85.7% rank-8 value quoted in Section 4.3 comes from a different, class-mean protocol (projecting $\boldsymbol{\mu}_{\text{res}} - \boldsymbol{\mu}_{\text{comp}}$ onto a PCA basis fit on resistant states and replacing the state with $\boldsymbol{\mu}_{\text{comp}}$ plus the projection), so it is not directly comparable to the 81.0% here.

Table 5: Rank-k PCA subspace patching at the peak layer (Qwen3-4B L24; Qwen3-14B L28, Qwen3-32B L58), per-stimulus protocol. The 80% threshold is crossed at k=8 for Qwen3-4B and Qwen3-14B and at k=64 for Qwen3-32B (2560, 5120, and 5120 hidden dimensions).

k	Qwen3-4B (L24)	Qwen3-14B (L28)	Qwen3-32B (L58)	Note
1	33.3%	54.3%	45.6%	rank-1 PCA
2	45.2%	56.8%	51.6%	
4	63.1%	74.1%	70.9%	
8	**81.0%**	**80.2%**	72.5%	**80% threshold (4B, 14B)**
16	89.3%	84.0%	79.1%	near-full recovery
64	90.5%	86.4%	**80.8%**	**80% threshold (32B)**
Full	90.5%	87.7%	81.3%	upper bound (no projection)

C SUBLAYER DECOMPOSITION

Each transformer layer contains two sublayers (multi-head attention and an MLP) connected by a residual stream that carries the input forward. To determine which component is responsible for the causal signal, we run three patching variants at each of seven layers: (a) *full*, replacing the entire residual-stream output; (b) *attn-only*, replacing only the attention sublayer output; and (c) *MLP-only*, replacing only the MLP sublayer output.

If the causal signal were generated *within* the target layer's attention or MLP, then patching those sublayers alone should recover most of the full-patch effect. The result (Table 6) is clear: at L24, neither sublayer alone accounts for more than 24% of the full 90.5% effect. The signal is not computed locally. Instead, it arrives at L24 already encoded in the *incoming residual stream*, having been built up progressively across preceding blocks.

This finding constrains the causal interpretation: L24 is a *leverage point*, not a *computation site*. The compliance representation is built up progressively across the preceding blocks and arrives encoded in the residual stream entering L24; L24 itself does not freshly compute the compliance outcome. Patching at L24 is therefore best understood as intercepting a signal in transit rather than overwriting a local decision unit. This is consistent with the broader late-layer localization pattern: the representation is transformed, not created, at the peak layer.

D INDIRECT INJECTION FULL RESULTS

For indirect injection, the adversarial payload is buried inside a simulated knowledge-base retrieval block in the user turn rather than stated directly. We constructed 600 stimuli (300 topics $\times$ 2 prefix variants) and measured a 25.0% natural compliance rate (150/600), compared with 28% on the multi-template direct corpus. Framing matters in both directions rather than uniformly reducing potency: the support-ticket framing of the 60-stimulus indirect pilot below yields 73% (44/60), so we make no general claim about indirect framing and potency.

Table 6: Sublayer decomposition at key layers (Qwen3-4B). At L24 (peak), neither attention nor MLP alone accounts for more than 24% of the full-patch effect (90.5%), identifying the *incoming residual stream* as the primary causal carrier.

Layer	Full	Attn-only	MLP-only	Primary site
L1	3.6%	1.2%	1.2%	Neither
L8	7.1%	1.2%	2.4%	Neither
L16	15.5%	6.0%	9.5%	MLP
L24	**90.5%**	21.4%	21.4%	**Incoming residual**
L28	89.3%	7.1%	21.4%	MLP + residual
L32	85.7%	10.7%	10.7%	Incoming residual
L36	88.1%	7.1%	17.9%	Joint/residual

Indirect-own layer sweep. Table 7 reports the sparse layer sweep on the N_c=150 compliant stimuli, patching the global resistant centroid of the indirect corpus itself. The raw flip rate rises late and peaks at L36 (43.3%), but at L36 a norm-matched random vector also flips 43.3% of the targets (compliant mean 43.3%, benign mean 30.0%), so the global indirect-own donor carries no class-specific signal. With 300 retrieval topics in the resistant pool, its centroid sits near the grand mean of the corpus, and its displacement from the compliant state is dominated by topical variation rather than by the compliance contrast.

Table 7: Indirect injection with the global indirect-own centroid (Qwen3-4B, 600-stimulus corpus, N_c=150). The raw rise is late, but at L36 the norm-matched random control also flips 0.433.

Layer (depth)	Flip rate
L1 (3%)	0.087
L8 (22%)	0.133
L16 (44%)	0.160
L24 (67%)	0.287
L28 (78%)	0.280
L32 (89%)	0.400
L36 (100%)	**0.433**

Clean-donor transfer and operator robustness. The dilution above is a property of the indirect-*own* donor, not of the target layer. Transferring a *clean* donor — the direct-injection resistant/compliant contrast — onto the N_c=44 compliant targets of the 60-stimulus indirect pilot corpus (whose indirect-own donor pool holds only 16 resistant states) recovers a 95.5% flip rate at L24 (median generation perplexity 1.7, i.e. coherent refusals), versus 15.9% (L24) and 6.8% (L36) for the diluted indirect-own centroid. Under this clean donor the effect is operator-robust (Table 8): a full centroid, a 2×-scaled contrastive direction, and a rank-8 pooled subspace flip 95.0%, 92.5%, and 85.0% of cases respectively (N_c=40 at L24, a second regeneration of the same corpus), whereas a rank-8 subspace built from resistant states alone reaches only 55.0%. The compliance geometry at the late layer is therefore intact for indirect injection; the original global centroid simply lacked a donor that isolates the compliance contrast from topical variation.

E Multi-turn Escalation

We test whether late-layer localization persists when the injection arrives after a benign conversational turn. We construct 360 two-turn stimuli using two softer injection templates, three benign warm-up queries, and 60 topics. Turn 1 contains a normal AcmeCorp support query, while Turn 2 introduces the injection. We collect hidden states from the full conversation and apply the same last-token patching protocol as in the single-turn setting.

Of the 360 stimuli, 62 (17.2%) elicit compliance. Table 9 shows that the causal profile remains late. Flip rates are negligible at L1, rise sharply from L16 to L24, and peak at L24 with 87.1%, close to the 91.7% single-turn peak.

Table 8: Indirect injection with a clean direct-injection donor at L24 (Qwen3-4B). The late-layer effect recovers to direct-injection levels and is robust across operators; the diluted indirect-own centroid is shown for comparison. Targets: 60-stimulus indirect pilot corpus (N_c=40 for the operator rows, N_c=44 for the indirect-own rows).

Donor / operator	Flip rate
Clean direct donor, full centroid	0.950
Clean direct donor, scaled direction (2×)	0.925
Clean direct donor, rank-8 pooled subspace	0.850
Clean direct donor, rank-8 resistant-only	0.550
Indirect-own centroid (L24)	0.159
Indirect-own centroid (L36)	0.068

Table 9: Multi-turn causal sweep on Qwen3-4B (N_c=62). The peak remains at L24, matching the single-turn setting.

Layer	Flip rate	Depth
L1	0.016	2.8%
L16	0.113	44%
L20	0.661	56%
L24	**0.871**	**67%**
L28	0.855	78%
L32	0.742	89%
L36	0.484	100%

These results show that a benign conversational prefix does not shift the causal region. Whether the injection appears in the first or second user turn, causal leverage remains concentrated in the same late block.

F Public-Benchmark Generalization

F.1 Benchmark-matched donors (deepset, neuralchemy)

This experiment addresses two questions: (1) does causal leverage concentrate in the late block for naturalistic (non-synthetic) attacks, and (2) does the lower flip rate observed with AcmeCorp donors (18.9%) reflect a donor-mismatch confound? We answer both with a dense 19-layer sweep using donors drawn from the benchmark distribution itself.

Design. We sampled 300 injections from two public benchmarks (`deepset/prompt-injections` and `neuralchemy/Prompt-injection-dataset`), split deterministically: the first 150 for calibration, the last 150 for testing. Resistant centroids at each sweep layer were computed from the 55 calibration-resistant stimuli (N_r=55, compliance rate 63%, N_c=95 calibration-compliant). These benchmark-matched centroids were then applied to the 90 test-compliant stimuli across all 19 sweep layers ($\ell \in \{1, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36\}$).

Results. Results are shown in Table 10.

Interpretation. The sweep resolves both questions cleanly. First, *causal localization generalizes to naturalistic attacks*. The null region spans L1–L18 (all ≤3.3% across ten consecutive layers), and the curve rises in the final third, peaking at L24 (67% depth), the same relative location as in the AcmeCorp experiments. Second, *the lower absolute flip rate is not explained by donor mismatch*. Benchmark-matched donors actually produce a *lower* peak flip rate (12.2%) than the cross-distribution AcmeCorp donors (18.9%), ruling out mismatch as a confound. The resistant signal remains class-specific: +11 pp above norm-matched random (12.2% vs. 1.1%), confirming that a compliance geometry exists at L24 for naturalistic phrasings. The roughly 7.5× reduction in absolute flip rate relative to the synthetic setting (12.2% vs. 91.7%) reflects a weaker and less

Table 10: Dense 19-layer causal sweep on naturalistic benchmark injections with benchmark-matched donors. Checkmark = null region (flip rate $< 5\%$). Layers L1–L18 (first 50% of depth) form a clean null region; the curve rises in the final third and peaks at L24 (67% depth), replicating the AcmeCorp curve shape at roughly $7.5\times$ lower absolute rates.

Layer	Depth	Flip rate	Null?
L1	3%	1.1%	✓
L2	6%	0.0%	✓
L4	11%	0.0%	✓
L6	17%	0.0%	✓
L8	22%	0.0%	✓
L10	28%	1.1%	✓
L12	33%	3.3%	✓
L14	39%	3.3%	✓
L16	44%	3.3%	✓
L18	50%	3.3%	✓
L20	56%	8.9%	
L22	61%	8.9%	
L24	**67%**	**12.2%**	**(peak)**
L26	72%	10.0%	
L28	78%	10.0%	
L30	83%	10.0%	
L32	89%	10.0%	
L34	94%	10.0%	
L36	100%	10.0%	
3-condition controls at L24 (peak):			
Resistant mean	—	12.2%	
Compliant mean	—	3.3%	
Norm-random	—	1.1%	

consistently aligned mean-difference signal across diverse real-world phrasings, not a failure of causal localization.

F.2 Additional public benchmarks: BIPIA and Gandalf

Beyond the deepset/neuralchemy phrasings of Appendix F, we replicate the causal sweep on two further public prompt-injection datasets under the AcmeCorp system prompt: `microsoft/BIPIA` (indirect injections embedded in email/web/table/summarization/code QA; we extract the $[N]$ attack payloads and present each directly in the user turn, so this tests BIPIA attack content rather than its indirect threat model) and `Lakera/gandalf_ignore_instructions` (crowd-sourced, human-written "ignore-your-instructions" attacks, used as released). For each layer we patch with the dataset's own resistant mean and compare against a *same-layer* norm-matched random control ($\boldsymbol{\mu}_{\text{comp}} + \text{unit-random} \cdot \|\boldsymbol{\mu}_{\text{res}} - \boldsymbol{\mu}_{\text{comp}}\|$). The flip-minus-random gap, not the raw flip rate, is what distinguishes a class-specific effect from generic perturbation.

Table 11 shows both datasets peak at L24, where the class-specific gap is largest (BIPIA $+0.57$, Gandalf $+0.34$) and collapses to ≈chance by L36, so the elevated raw flips in the very last layers are mostly generic perturbation, and L24 is the true class-specific peak. The raw flip rates at L24 (0.66–0.70) exceed the naturalistic `deepset/neuralchemy` rate ($\sim$12%, Appendix F) because these datasets use narrower, more templated phrasing; the causal *location* is identical. This is consistent with our central claim that the late-layer location is distribution-robust while the absolute magnitude tracks phrasing diversity.

G Agentic Prompt Injection (InjecAgent, AgentDojo)

To probe threat models beyond single-turn role conflict, we evaluate two agentic benchmarks in ReAct format: **InjecAgent**, with poisoned tool outputs targeting data theft or direct harm, and **AgentDojo**, with attacker goals across banking, Slack, travel, and workspace tasks. We use Mistral-7B as the

Table 11: Causal sweep on two public benchmarks under the AcmeCorp system prompt (Qwen3-4B; BIPIA payloads in the user turn, N_c=23; Gandalf N_c=44) Each cell is flip rate / same-layer norm-matched random control. Both peak at L24 with the largest gap over random.

Layer (depth)	BIPIA (flip / rand)	Gandalf (flip / rand)
L1 (3%)	0.043 / 0.043	0.023 / 0.068
L8 (22%)	0.130 / 0.130	0.091 / 0.068
L16 (44%)	0.304 / 0.348	0.205 / 0.159
L20 (56%)	0.435 / 0.391	0.568 / 0.386
L24 (67%)	**0.696 / 0.130**	**0.659 / 0.318**
L28 (78%)	0.652 / 0.261	0.636 / 0.614
L32 (89%)	0.652 / 0.435	0.614 / 0.545
L36 (100%)	0.609 / 0.609	0.568 / 0.568

primary testbed because it exhibits sufficient attack success: $36/62$ InjecAgent and $10/24$ AgentDojo stimuli trigger the attacker-requested action.

The attack is decodable, but single-token causal leverage is not concentrated. A linear probe separating injected from benign tool results (matched pairs differing only in the tool output) reaches attack-*presence* AUROC $= 1.00$ from the first layer on Qwen3-4B, Qwen3-14B, and Mistral-7B. Presence, however, is not the compliance decision: a within-agentic *compliance* probe (compliant vs. resistant rollouts) peaks at only AUROC 0.72 (InjecAgent) and 0.54 (AgentDojo). Causally, patching the resistant mean at the prompt-final token *or* at the commitment token (where the attacker action is emitted) does not exceed a same-layer norm-matched random control at any layer (e.g. InjecAgent L16: commit 0.28 vs. random 0.31).

Interpretation. The clean early-decode / late-cause dissociation of the single-step setting *widens* rather than breaks: the attack is fully present in the residual stream from layer 1, yet the behavioral decision is produced across multiple generated steps, so no single-token intervention controls it. This is consistent with our central claim that representational availability does not imply a localized causal decision. Localizing the distributed agentic decision (e.g. multi-token or multi-step patching) is future work.

H Additional Models: Reasoning, MoE, and Training Stage

H.1 Reasoning and Mixture-of-Experts

Reasoning (thinking) mode. Re-running Qwen3-4B with its native chain-of-thought (`enable_thinking=True`) leaves the attack decodable and preserves a late causal peak at L31 (86% depth), but the absolute flip rate is much weaker than in the non-thinking setting (peak 0.21). When the model emits an explicit reasoning trace, the compliance commitment appears to spread across generated reasoning tokens, so a single prompt-final patch is no longer the natural intervention point.

Mixture-of-Experts. On Qwen3-30B-A3B (48 layers, $\sim$3B active parameters per token) the causal signal is much noisier: the peak flip rate is 0.06 at L21, barely above the same-layer random control (0.04). We therefore make no strong claim that the dense single-token bottleneck carries over to sparse MoE routing, and whether expert routing distributes the compliance computation is an open question. Both regimes indicate that the single-token causal intervention is strongest for dense, non-thinking models.

H.2 Training-stage attribution: base vs. instruct

To ask *where in training* these representations arise, we compare Qwen3-4B-Base (pre-training only) and Qwen3-4B (instruction-tuned) under identical prompts and formatting (Table 12). The injection representation is already fully present after pre-training: a linear-SVM probe (5-fold AUROC on the 300 headline injections vs. 60 ordinary benign queries) separates injection from benign with AUROC $= 1.00$ at L1 in the base model, as it does in the instruct model on the same set, so instruction tuning does not create decodability from scratch. This is the same headline set as the detector of

Section 7 (where the L1 AUROC of 0.888 is measured against hard negatives, not ordinary benign queries) and a larger set than the 30-vs-30 probe pilot behind the 80% accuracy of Section 4.1. What post-training changes is behavior and geometry: it lowers the compliance (attack-success) rate from 0.65 to 0.28 and raises the late-layer contrastive-direction norm at L32 by 74% (72.5 $\rightarrow$ 126.0). In short, pre-training builds the representation; post-training suppresses compliance and sharpens the late-layer direction into behavioral control.

Table 12: Base-vs-instruct comparison for Qwen3-4B under identical prompts. Decodability is already maximal after pre-training; instruction tuning suppresses compliance and sharpens the late-layer direction.

Model	Attack decodable at L1	Direction norm at L32	Compliance (ASR)
Qwen3-4B-Base (pre-training only)	1.00	72.5	0.65
Qwen3-4B (instruct)	1.00	126.0	0.28

I External detector baseline and adaptive evasion

Baseline. We compare the L24 activation probe against an off-the-shelf text classifier, `ProtectAI/deberta-v3-base-prompt-injection`, on the hard-negative set of Section 7. The L24 probe reaches AUROC 0.993 versus 0.709 for the text classifier, and it reuses a prefill activation (one PCA projection, standardization, and a linear readout) rather than a second model forward pass.

Adaptive attacker. Because a deployed monitor's location may become known, we stress-test with a white-box adversary: universal GCG optimization of an adversarial suffix that minimizes the L24 detector score while preserving compliance. Table 13 reports recall at a 5%-FPR threshold. The suffix roughly halves L24 recall (1.00 $\rightarrow$ 0.48) while the model stays compliant, as expected for any single-layer monitor. Crucially, the same suffix barely transfers to the text classifier (0.625 $\rightarrow$ 0.600): the evasion is specific to the L24 activation geometry, not a generic prompt-injection jailbreak. We therefore treat detection as mechanistic corroboration and a complementary lightweight signal, not a standalone defense.

Table 13: Detector recall at a 5%-FPR threshold, before and after a white-box GCG suffix optimized against the L24 score (Qwen3-4B). The suffix targets L24 activations and does not transfer to the text classifier.

Detector (recall @ 5% FPR)	Plain injection	Adaptive GCG suffix
L24 activation probe	1.000	0.475
ProtectAI DeBERTa-v3	0.625	0.600