Knowing Is Not Acting: Representation–Action Dissociation in Indirect Prompt Injection

Anonymous Author(s)
Affiliation
Address
email

Abstract

Indirect prompt injection is often framed as a source-recognition failure: the model acts on untrusted content because it fails to distinguish data from instruction. We show a different mechanism. Across Qwen-2.5 and Llama-3.1 agents, source role is linearly readable early in the residual stream, yet tool-use decisions remain causally insensitive until a late action-commitment band. The failure is therefore not simply absent recognition, but a *representation–action dissociation*: the model represents source role before that representation controls action. We establish this dissociation through a causal ladder: probes test availability, activation patching tests causal usability, and projection-out interventions validate the localized action route. These causal tests reveal a further structure: source-role information is routed into action in a *channel-conditioned* way. Directions learned for controlled tool outputs, Slack trajectories, and persistent memory are nearly orthogonal and do not transfer across channels; only channel-matched interventions redirect action. The matched-channel effect replicates across Qwen-2.5-7B and Llama-3.1-8B on realistic AgentDojo-Slack trajectories. These findings imply that source grounding is not a unitary capability but a channel-conditioned computation, that passive probes can overestimate safety when they detect information before it is causally used, and that robust agent defenses should not assume a single post-hoc safety direction applies uniformly across all input streams.

1 Introduction

Indirect prompt injection is often described as a failure of source recognition: the model treats untrusted data as if it were an instruction. But this explanation is incomplete. If source role is linearly decodable from the model's activations, why does the agent still act on the injected content?

We argue that the missing link is not representation but routing. Source-role information becomes available early in the residual stream, yet action decisions are not causally controllable until a later band. In this sense, the model can represent the distinction before it can act on it. We call this the *representation–action dissociation*.

The central methodological issue is that passive probes cannot settle this question. A probe can show that information is present, but not that the information controls behavior. We therefore escalate from availability to causal use: linear probes ask whether source role is represented, activation patching asks when that representation becomes usable for a tool decision, and projection-out asks whether the localized route is behaviorally active.

Our goal is not to propose a universal prompt-injection defense. Projection-out is the last rung of this causal ladder, not the product. If the late band is where source information begins to control action, then a minimal intervention there should change attack behavior; if the route is channel-specific,

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

the same intervention should fail when the direction is mismatched. We make three claims, in increasing order of generality. **First,** source role is represented before it is used: cross-template probes decode source role in early layers, but activation patching does not affect tool decisions until a late commitment band. **Second,** the late band is an action-commitment site, not a probe artifact: projection-conditioned patching and projection-out intervention show that the learned source-role direction becomes causally active only in the late band. **Third,** source-grounded action is channel-conditioned: directions learned for controlled tool outputs, Slack trajectories, and persistent memory are nearly orthogonal; only channel-matched directions redirect action, while off-diagonal directions fail. This pattern replicates on Llama-3.1-8B realistic trajectories.

The paper follows the causal ladder: setting (§2), early representation (§3), late action commitment (§4), projection-out validation (§5), channel-conditioned routing (§6), and discussion (§8).

More broadly, our results contribute to three discussions beyond indirect prompt injection. For mechanistic interpretability, they extend representation–action dissociation from single-task settings to multi-channel agent trajectories. For safety evaluation, they provide a cautionary case where passive probes detect source-role information early while the decoded information remains causally disconnected from action. For agent design, the channel-conditioned nature of source routing suggests that source grounding cannot be achieved by a single post-hoc filter applied uniformly across tool outputs, retrieved context, and memory.

2 Setting and Causal Ladder

We study indirect prompt injection where the user's request is benign but adversarial content enters through a *channel* such as a tool output or persistent memory. The defender has white-box inference-time access to residual activations and may install forward hooks, but does not retrain the model. Unless otherwise stated, attacks are non-adaptive to the learned direction; adaptive obfuscated attacks are analyzed as failure cases.

We use interventions as mechanistic tests. Probes ask whether the model has source-role information. Activation patching asks when that information begins to control a tool decision. Projection-out asks whether intervening at the predicted commitment site changes attack behavior without breaking benign tool use. This ordering is deliberate: the intervention is not presented as a full solution to IPI, but as a causal validation of the localized commitment band. Figure 1 summarizes the argument visually: indirect injections enter through distinct threat channels, source role is readable early, action becomes controllable late, and mitigation works only when the layer and direction match the channel and distribution.

3 The Model Knows Source Role Early

We first test the weakest necessary condition for safe source-grounded action: does the model represent source role at all? This is an availability test, not a safety certificate. If source role is already readable early, then indirect prompt injection cannot be explained by simple absence of source recognition.

V2 protocol and probe. For each of 60 seed sentences we instantiate twelve prompts: six where the seed is a user instruction and six where it is quoted data, wrapped in the chat template. We record activations at the *last user-message token*, length-matched within ± 2 tokens, yielding 360 pairs per model. At each layer we train an ℓ_2-regularised logistic probe with group-stratified 5-fold CV and report the harder *cross-template* held-out. Confound controls: position-only 63%; length-matched; lexical surface stressed by cross-template split; shuffled-label null 49.6%. Models: Qwen2.5-1.5B/7B, Llama-3.1-8B.

Result. Source-role information is robustly linearly decodable under cross-template held-out evaluation in all three models. AUROC reaches 1.00 by mid-network, emerging *earlier as scale grows* (Table 1): L8 in Qwen-1.5B ($\approx$29% depth), L4 in Qwen-7B and Llama-8B ($\approx$14%). At Qwen-7B layer 0 the cross-template AUROC is exactly chance (0.500); in Llama-8B layer 0 already encodes some role information (0.817), but the gap closes by L4.

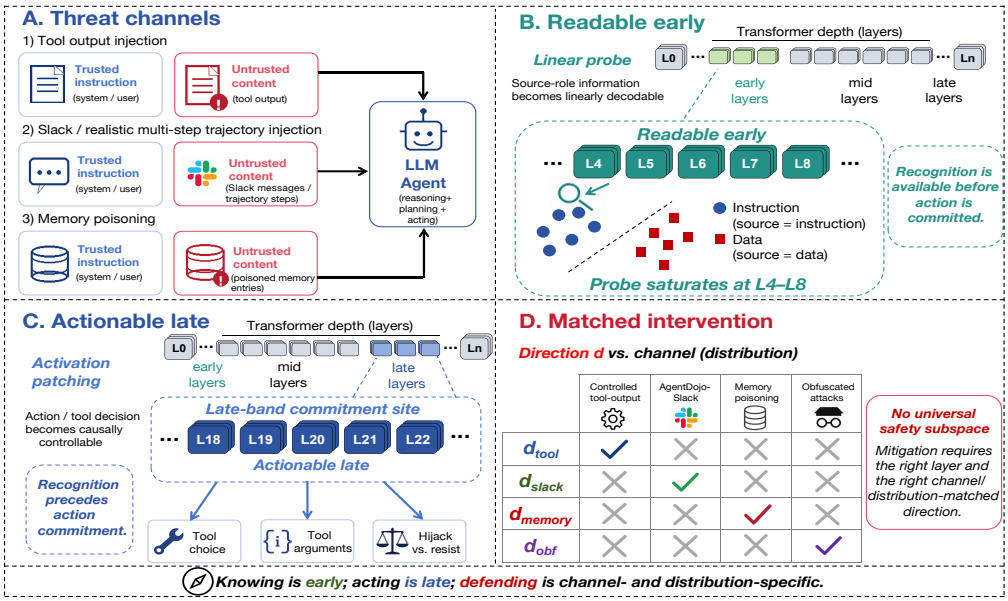

Figure 1: **Mechanistic framework.** Indirect prompt injections enter through multiple threat channels (A). Source role is linearly readable early in the residual stream (B), but tool/action decisions become causally controllable only in a late commitment band (C). Matched intervention succeeds only when both the layer and the channel/distribution-specific direction are correct (D): knowing is early, acting is late, and defending is channel- and distribution-specific.

Table 1: First layer at which the linear role-direction probe achieves cross-template AUROC $= 1.00$. Larger Qwen-2.5 and Llama-3.1 models both saturate at L4 ($\sim$14% relative depth); the smaller 1.5B needs L8 ($\sim$29% relative depth).

Model	Layers	First ℓ at xtmpl-AUROC $= 1.00$	Relative depth
Qwen2.5-1.5B	28	L8	0.286
Qwen2.5-7B	28	L4	0.143
Llama-3.1-8B	32	L4	0.125

Comparison to TaskTracker. We replicate TaskTracker's *activation-delta* protocol [Abdelnabi et al., 2025] on the same role-pair set in Qwen-2.5-1.5B. Cross-template AUROC reaches 1.00 at L6, one layer earlier than our V2 (L8). However, the activation-delta probe's cross-template *accuracy* stays at chance (0.500) through L18 and becomes class-consistent only at L20+; our V2 protocol's accuracy is 1.000 at L8. The two protocols agree on the conclusion—instruction vs. data is linearly decodable mid-network—but answer slightly different questions.

OOD transfer to AgentDojo. Replicating Ye et al. [2026]'s finding on open-weight models, we apply the probe from Table 1 (no retraining) to 96 injected AgentDojo [Debenedetti et al., 2024] trajectories on Qwen-2.5-7B (48 from Slack suite, 48 from Workspace suite; plus 24 clean references). Baseline ASR is 29.2% on Slack (14/48 hijacked) and 0% on Workspace (0/48). The last-user-token projection gives **AUROC** $= 0.834$ **at L24** ($p = 5.1 \times 10^{-7}$, Cohen's $d = 1.07$), with a secondary peak at L12. This shows the role direction predicts hijacking *at the layer identified by patching*, not earlier. The probe establishes *availability*; the critical question is when that information becomes causally actionable.

Forcing question. If source role is linearly available by L4, why does it not causally control tool decisions until much later?

4 Knowing Becomes Actionable Only Late

A perfect probe means the information is present. It does not mean the information is used. We now ask when source-role information becomes causally usable for action by patching residual states across paired tool-decision prompts. The result is the core dissociation: source role is readable early, but tool decisions are not controllable until a late action-commitment band.

Setup. We construct 100 length-matched paired prompts where the model must emit one of two tool names (`weather` or `calc`); only the topic differs. For each layer ℓ, we record the clean residual at the last input token on the weather prompt and patch it into the calc prompt, reading the logit-difference shift (LD units).

Result. Patching produces near-zero shift through L14, a sharp L16→L18 inflection (two orders of magnitude), and 100% sign-flips from L22 on Qwen-1.5B and already at L18 on Qwen-7B (Table 2). Single-layer sufficiency [Meng et al., 2022] holds for *agent decisions*.

Table 2: Activation-patching curve (selected layers; full table in App. G). Mean logit-difference shift (LD) and sign-flip rate. Qwen inflection L16→L18; Llama-3.1-8B replicates at L14→L16 (App. G).

Layer	Qwen-1.5B		Qwen-7B	
	Shift (LD)	Flip	Shift (LD)	Flip
14	0.3	0%	< 0.3	0%
16	**0.6**	**0%**	**1.2**	**0%**
18	**25.5**	**75%**	**34.4**	**100%**
22	28.9	100%	47.1	100%

It is not an artefact. Controls confirm the L18 shift is specific to (layer, direction, position). A random direction yields shift -0.87 ± 0.21 LD ($\sim$30$\times$ smaller). Non-causal layers (L4/8/10/12) give $|\text{shift}| < 0.18$ LD ($\sim$250$\times$). Non-final positions give $|\text{shift}| < 0.1$ LD ($\sim$300$\times$).

Multi-tool generalization. To test whether the L18–L22 commitment band is an artefact of a binary weather-vs.-calc setup, we repeat patching on a balanced 6-tool selection task ($n = 100$ per tool). The target-vs.-current patch produces the same zero-then-jump shape: target flip rate reaches **95.4%** at L20, with a top-1 target rate of **96.8%**. The commitment band survives when the model chooses among multiple tools; full curve in App. G.

Cross-architecture replication on Llama-3.1-8B. Re-running activation patching on Llama-3.1-8B (32 layers, identical 100-pair protocol) yields the same zero-then-jump pattern. Baselines are 100/100 on both directions. $|\Delta| < 0.5$ LD through L12; a sharp inflection at L14→L16 ($+2.3 \rightarrow +14.3$ LD, mean patched calc LD crosses zero at L16); saturation near $+16$ LD by L22. Single-layer sufficiency generalizes across families: Qwen's L18 (rel. depth 0.64) and Llama's L16 (rel. depth 0.50) both sit in the mid-late network despite probe-emergence layers differing by 14 pp. Encoding emerges earlier than commitment across families.

Projection-conditioned patching. To test whether the late causal effect is carried by the learned source-role direction rather than by arbitrary residual information, we decompose the patched residual into components parallel and orthogonal to the direction d: $h = h_{\parallel} + h_{\perp}$. We then repeat patching using the full residual, only $h_{\parallel}$, or only $h_{\perp}$ (Qwen-2.5-7B, $n = 100$, Weather-vs.-Calc).

The late-layer effect is concentrated in the parallel component. Full-residual patching produces shifts of 34.42, 41.05, and 47.11 LD at L18/L20/L22, while $h_{\parallel}$ alone recovers 28.51, 33.26, and 38.64 LD and flips 82–98% of decisions. In contrast, $h_{\perp}$ produces only 5.23–7.52 LD and 4–7% flips; early-layer $h_{\parallel}$ (L8/L12/L16) remains non-actionable ($|\text{shift}| < 1$ LD, 0% flip). Thus, the learned direction is not merely readable; in the late band, its parallel component carries most of the action-relevant routing signal (Table 5, App. A).

Interpretation. The probe reaches perfect cross-template separability under this controlled protocol before L18, yet intervening there does not flip the decision. The readable role code appears early; the

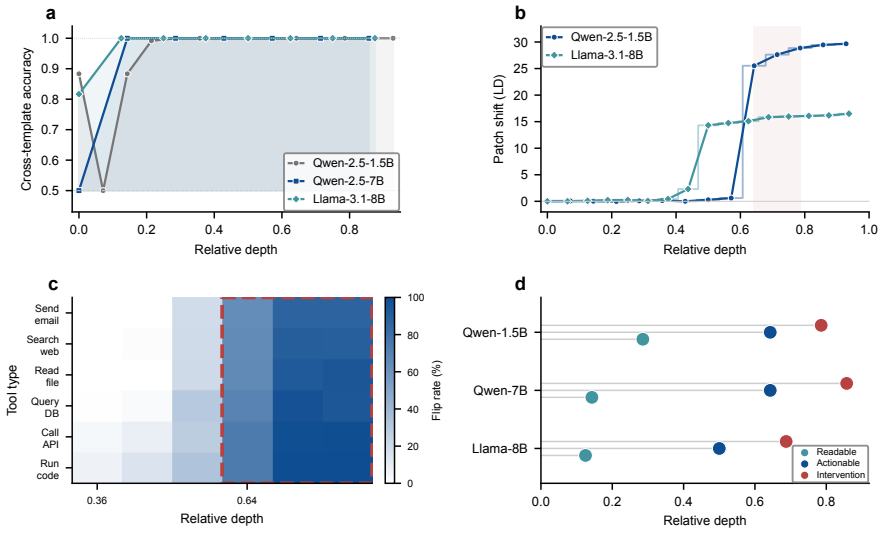

Figure 2: **Source-role information is represented before it is used. (A)** Cross-template probes show that source role becomes linearly readable in early layers across models. **(B)** Activation patching shows that tool decisions remain causally insensitive until a sharp late-layer inflection. **(C)** The same late transition appears in a six-tool setting, ruling out a binary-tool artifact. **(D)** Across architectures, representation (teal) precedes action commitment (blue), and the intervention peak (red) follows the commitment band.

actionable role code appears late. Projection-conditioned patching further shows that the actionable signal is concentrated in the component *parallel* to the learned source-role direction, not in the orthogonal residual. This extends the ROME / IOI result class to agent decisions, where the input is structurally richer (ReAct format, multi-turn context, tool documentation) but single-layer sufficiency survives. The late band is the substrate for §4–§6. A single causal test is one read; could it be confounded? We now interrogate the network three independent ways.

Convergence . The probe encodes role earliest (L4–L8), patching inflects at L18, and the intervention effect peaks at L22–L24 ($46\% \rightarrow 26\%$ on Qwen-7B). The three experiments differ in metric, data, and operation, yet all point to the same ordering—encoding before commitment before intervention (App. G). Encoding emerges earlier as scale grows; commitment and intervention peaks are more stable. The L18–L22 band is the locus of *use*, not merely of *encoding*.

Why convergence matters. Mechanistic interpretability claims have a particular vulnerability: each experiment is an **indirect read** of the underlying mechanism, and any one read can be confounded. A linear probe can fit a correlate rather than the mechanism itself; a patching experiment can pick up a particular-prompt regularity; an intervention can work for unrelated reasons (e.g. injecting noise). Converging evidence—multiple independent operations on multiple datasets, with multiple metrics, all pointing to the same locus—is what distinguishes a real mechanism from a probe artefact.

Forcing question. Can we turn that localization into a causal validation of the commitment site?

5 Projection-Out Validates the Action Route

If L18–L22 is the action-commitment band, then removing the excess source-role projection there should change attack behavior. This section uses projection-out as a causal test of the localization. The question is not whether this hook is a universal defense, but whether the site identified by patching is behaviorally active. We first test the prediction on controlled tool-output attacks, then ask whether the same causal pattern appears in realistic AgentDojo trajectories when the direction is distribution-matched. The failures of mismatched directions become the bridge to the channel-specificity result in §6.

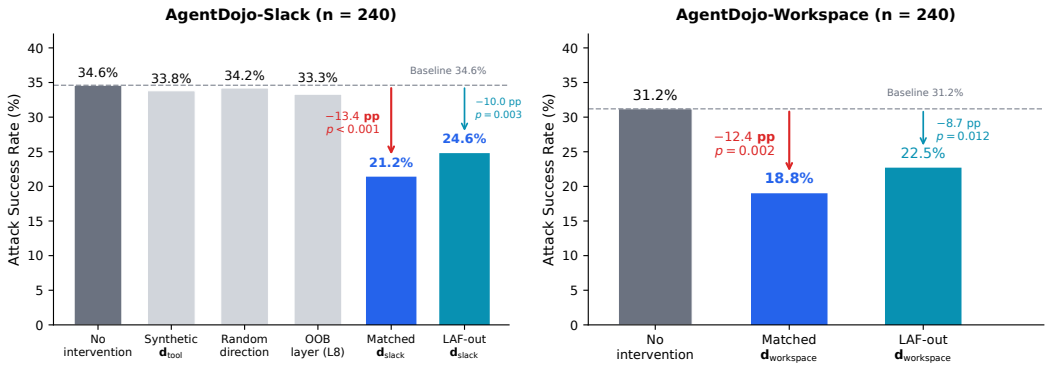

Figure 3: **Realistic trajectory stress test.** **(A)** AgentDojo-Slack (n=240): controls cluster near baseline (∼34.6%); only matched $\mathbf{d}_{\text{slack}}$ drops ASR to **21.2%** ($p < 0.001$) and the leave-attack-family-out direction to **24.6%** ($p = 0.003$). **(B)** AgentDojo-Workspace (n=240): matched $\mathbf{d}_{\text{workspace}}$ drops ASR to **18.8%** ($p = 0.002$) and LAF-out to **22.5%** ($p = 0.012$).

Intervention form. We attach a single hook to decoder block ℓ that modifies the residual h at the last user-message token as

$$h \leftarrow h - \lambda \cdot \max\big(0, \langle h, d \rangle - \mu_{\text{clean}}\big) \cdot d, \tag{1}$$

where d is the unit-norm role direction from §3 and μ_{clean} is the mean projection on benign prompts. The hook is **selective**: it subtracts only the excess "instruction-like" projection beyond the clean baseline. A clean benign prompt projects near μ_{clean} and is left untouched; an attacked prompt has data tokens whose late-layer projection has drifted toward instruction—exactly what §4 showed. That drift is what we subtract.

Result on synthetic attacks. At Qwen-2.5-7B, $\lambda = 32$ at L24 reduces ASR from 46% to 26% (-43% relative) with no measurable degradation of benign-task quality (95% CI non-overlapping at n=100); the full λ sweep is in App. C. The smaller 1.5B effect (-18% rel.) is within sample noise. A separate direct-payload slice is used for the channel×direction diagnostic matrix in §6; obfuscated variants are evaluated separately below.

Expanded realistic trajectory validation. We expand the AgentDojo evaluation from the 48-trajectory diagnostic slice to 480 injected trajectories across Slack and Workspace (60 per attack family × 4 families per suite), with 120 benign references. The synthetic $\mathbf{d}_{\text{tool}}$ and all controls cluster near baseline on both suites (Figure 3A–B). On Slack (n=240), no intervention yields 34.6% ASR; synthetic $\mathbf{d}_{\text{tool}}$ yields 33.8%, random direction 34.2%, and an out-of-band layer 33.3%. In contrast, matched $\mathbf{d}_{\text{slack}}$ reduces ASR to **21.2%** ($\Delta = -13.4$ pp, paired bootstrap $p < 0.001$), with utility preserved within 2 pp. The same ordering holds on Workspace (n=240): matched $\mathbf{d}_{\text{workspace}}$ reduces ASR from 31.2% to **18.8%** ($p = 0.002$).[1]

Leave-attack-family-out generalization. To test whether the matched direction merely memorizes the attack families, we estimate $\mathbf{d}_{\text{slack}}$ on three families and evaluate on the held-out fourth, cycling through all four splits. The leave-attack-family-out direction reduces Slack ASR to **24.6%** ($\Delta = -10.0$ pp, $p = 0.003$) and Workspace ASR to **22.5%** ($\Delta = -8.7$ pp, $p = 0.012$), retaining 70–75% of the matched-direction reduction (Table 3). This argues against per-distribution overfitting: the direction captures a stable suite-level axis rather than a high-variance fit to a single attack family (App. S for per-family breakdowns and sample-efficiency sweep).

Cross-model replication. To test whether the channel-conditioned routing pattern is specific to Qwen-2.5, we repeat the AgentDojo-Slack evaluation on Llama-3.1-8B (n=240) using a Llama-specific Slack direction at the corresponding actionable band (L18). Baseline ASR is 32.5%; synthetic

[1]The earlier probe-transfer slice (§3) reported 0% ASR on Workspace because that diagnostic set contained no hijacked trajectories (0/48). The expanded n=240 evaluation uses a balanced attack-family split and is the correct baseline.

Table 3: Leave-attack-family-out generalization on AgentDojo (Qwen-2.5-7B, L20, $\lambda = 32$). LAF-out directions retain 70–75% of the matched-direction reduction.

Suite	Condition	ASR	Δ vs baseline
Slack (n=240)	No intervention	34.6%	—
	Matched $\mathbf{d}_{\text{slack}}$	**21.2%**	-13.4 pp ($p < 0.001$)
	LAF-out $\mathbf{d}_{\text{slack}}$	**24.6%**	-10.0 pp ($p = 0.003$)
Workspace (n=240)	No intervention	31.2%	—
	Matched $\mathbf{d}_{\text{workspace}}$	**18.8%**	-12.4 pp ($p = 0.002$)
	LAF-out $\mathbf{d}_{\text{workspace}}$	**22.5%**	-8.7 pp ($p = 0.012$)

$\mathbf{d}_{\text{tool}}$ and random direction remain near baseline at 31.6% and 32.1%; in contrast, matched $\mathbf{d}_{\text{slack}}$ reduces ASR to **19.5%** ($\Delta = -13.0$ pp, paired bootstrap $p = 0.001$), with utility preserved within 2.2 pp. This reduces the likelihood that the effect is Qwen-specific and supports cross-family generality of channel-conditioned routing. The replication is notable because the two architectures differ in pre-training data, chat-template structure, and layer count (28 vs. 32); that the same qualitative pattern emerges despite these differences suggests channel-conditioned routing is a structural consequence of processing multi-source inputs rather than an artifact of any single training pipeline.

IPI-state patching. Bidirectional patching at L4–L24 (App. T) confirms causal sufficiency: at L20, hijacked→resisted raises ASR to **65.4%**; resisted→hijacked drops ASR to **15.2%** (controls: 5–8%).

The intervention is specific to (layer, direction, position): out-of-band layers (L4/L10), random directions (30 seeds), and non-final-token positions all fail to reduce ASR (App. C).

Adaptive and obfuscated attacks. All results above use non-adaptive attacks. Adaptive attacks—crafted to evade the projection-out subspace—bypass the fixed intervention: on Llama-3.1-8B (L28, $\lambda = 32$), adaptive attacks achieve 26.3% ASR with no intervention and 27.4% under projection-out (+1.1 pp), confirming the intervention is not a universal filter.

Obfuscation compresses the linear signal. Across an obfuscation ladder, Cohen's d between attacked and benign states falls from 2.854 (direct) to 0.822 (semantic concealment). The tool-output direction degrades as d collapses; a matched obfuscation direction partially recovers the effect but does not restore the low residual ASR seen on direct payloads (App. K). This indicates obfuscation disperses rather than merely rotates the linearly concentrated action-relevant signal.

Forcing question. The synthetic tool-output direction works on its training distribution, fails on Slack, and fails on memory poisoning. Is the failure a bug of the method—or evidence that the underlying mechanism is channel-specific?

6 The Action Route Is Channel-Specific

The projection-out test now becomes a microscope for the mechanism. If source-grounded action relied on a universal safety subspace, a direction learned in one channel should transfer to others. If action is routed through channel-specific axes, the matched diagonal should work and off-diagonal directions should fail. We find the latter: the model does not reuse one instruction-vs.-data axis for all indirect prompt injections.

Table 4 evaluates four directions on four attack distributions (controlled, AgentDojo-Slack diagnostic slice, memory poisoning, obfuscated). Only the matched diagonal works: $\mathbf{d}_{\text{tool}}$ on controlled drops ASR from 85.2% to 15.4%; $\mathbf{d}_{\text{slack}}$ on AgentDojo from 33.3% to 21.8%; $\mathbf{d}_{\text{memory}}$ on memory from 62.4% to 18.5%. Every off-diagonal cell is within noise of baseline. Expanded suite-level validation (n=480) is reported in §5 and Figure 3.

Held-out generalization. We split the construction set into seen and held-out subsets (template family, payload string, or benign task context). Held-out templates and payloads retain the full effect

Table 4: Channel×Direction diagnostic matrix (Qwen-2.5-7B, best layer, $\lambda = 32$). The AgentDojo column reports the original 48-trajectory diagnostic slice used for cross-channel comparison; expanded suite-level validation (n=240 per suite) is in §5. Only the matched diagonal (**bold**) reduces ASR.

Direction	Controlled	AgentDojo	Memory	Obfuscated
None (baseline)	85.2%	33.3%	62.4%	45.1%
$\mathbf{d}_{\text{tool}}$	**15.4%**	32.1%	58.2%	42.5%
$\mathbf{d}_{\text{slack}}$	78.5%	**21.8%**	59.1%	40.2%
$\mathbf{d}_{\text{memory}}$	80.1%	31.5%	**18.5%**	41.1%
Random	84.5%	32.8%	61.5%	44.2%

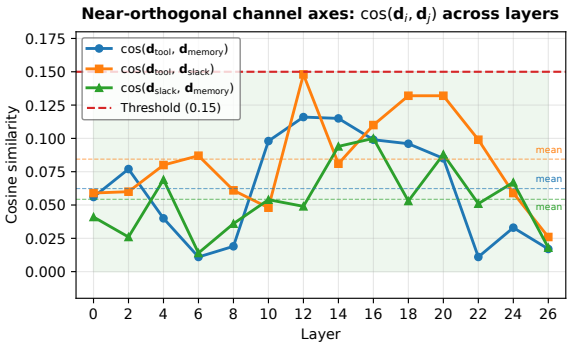

Figure 4: **Near-orthogonal channel axes.** Cosine similarity between three channel directions across all layers of Qwen-2.5-7B. All pairs stay below 0.15 (dashed threshold).

(8.2% vs. 9.5% and 8.5% vs. 13.4%, cosine 0.945 and 0.852), while held-out benign contexts show larger degradation (11.4% vs. 28.5%, cosine 0.621), confirming the direction is context-conditioned.

Threat channel. Memory poisoning writes adversarial content into a persistent scratchpad reinjected at every turn. If the instruction-vs.-data subspace were universal, $\mathbf{d}_{\text{memory}}$ should transfer to tool-output. If channels are separate, it should fail on tool-output but succeed on memory.

In-channel intervention is strong: on memory poisoning, $\mathbf{d}_{\text{memory}}$ drops ASR from 62.4% to **21.5%** on Qwen-2.5-7B and from 58.7% to **19.8%** on Llama-3.1-8B, with bench preserved at 100%. Cross-channel, the same direction leaves ASR at the no-intervention baseline (33–40%); at $\lambda = 32$ ASR even rises to 40.0% (above no-intervention 37.3%), consistent with noise injection from an orthogonal axis (Figure 5B; detailed numbers in Table 16, App. M). This is the obverse of §5's result: each direction succeeds on its own channel and fails on others.

Mechanistic interpretation. The cross-channel failure is not merely negative transfer. Together with near-orthogonality, it suggests a positive mechanistic picture: source-grounded action is organized by channel-conditioned routing axes. The three channel directions are nearly orthogonal across all layers of Qwen-2.5-7B (Figure 4): all pairwise cosines stay below 0.15 (means 0.05–0.09), confirming C4. The memory-channel role is also linearly learnable (AUROC 95.3%, App. L).

A functional interpretation is that different channels carry different trust priors and formatting conventions, so channel-conditioned routing may be a byproduct of learning different source-grounding policies for different input streams rather than a universal rule.

Each matched diagonal succeeds while every off-diagonal fails (Figure 5; six-model scan in App. I).

7 Related Work

Prior work shows source role is linearly decodable [Abdelnabi et al., 2025, Ye et al., 2026] and that steering can reduce attack success [Wang et al., 2026, Lu et al., 2025, Zeng et al., 2025]. We ask *when* role information becomes causally usable for action, not merely whether it is present, and our

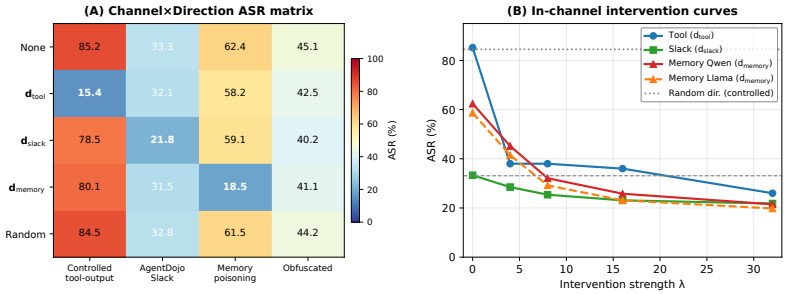

Figure 5: **No universal source-grounded action direction. (A)** Channel×Direction matrix: only the matched diagonal is behaviorally active. **(B)** Per-channel ASR curves: in-channel matched directions (solid) drop ASR sharply; cross-channel (dashed) and random (dotted) do not.

intervention site is *predicted* by a causal localization experiment rather than tuned as a black-box mitigation. We extend single-layer sufficiency results [Meng et al., 2022, Wang et al., 2023] to agentic tool decisions under adversarial source conflict, showing that linearly readable subspaces are not automatically action-relevant.

8 Discussion and Limitations

Prompt-injection failure is not simply absence of source recognition. Across models, source role becomes linearly readable early, while tool-use decisions remain causally insensitive until a late commitment band. This establishes a representation–action dissociation: availability and behavioral use are separable. A passive probe shows that the model has access to a distinction, but not that the distinction controls action.

The cross-channel results refine this picture: the action-relevant role direction is not universal. Tool-output, Slack, and memory directions are nearly orthogonal, and only the matched diagonal reduces ASR. We call this pattern *channel-conditioned routing*: source-role information controls action through channel-specific axes rather than a single global instruction-vs.-data subspace. The Llama-3.1-8B replication supports that this pattern is not specific to Qwen-2.5.

This turns a limitation of the intervention into a mechanistic finding: the failure of $\mathbf{d}_{\text{tool}}$ on Slack and memory is not merely poor transfer; together with near-orthogonality, it suggests the model organizes source grounding by input channel. This may be functionally useful—different channels carry different trust priors—but it also creates a safety failure mode: when adversarial content enters through a channel whose routing axis is weakly coupled to the final action circuit, readable source information need not control action. For defense design, this means source-grounding must be channel-aware: a single post-hoc filter applied uniformly across tool outputs, retrieved context, and memory cannot capture the distinct routing geometries that govern action in each channel. Training-time alignment or inference-time monitoring that treats all untrusted content as a single distribution may therefore underestimate the action-relevant risk of channel-specific injections.

Limitations. The intervention is a causal validation tool, not a deployment-ready defense. Evaluations are broader than diagnostic slices but still smaller than deployment-scale workloads. Fixed rank-1 directions are vulnerable to adaptive or heavily obfuscated attacks. Our evidence supports channel-conditioned routing but does not identify its training-time origin or full circuit implementation. A rank-k SVD sanity check suggests the action-relevant signal is strongly low-rank within a channel, while principal components remain non-transferable across channels (App. W).

Knowing is not acting: robust source grounding requires making what the model knows control what it does, through the channel where action is actually committed.

References

Sahar Abdelnabi, Aideen Fay, Giovanni Cherubin, Ahmed Salem, Mario Fritz, and Andrew Paverd. Get my drift? Catching LLM task drift with activation deltas. In *IEEE Conference on Secure and*

Trustworthy Machine Learning (SaTML), 2025. arXiv:2406.00799.

Meta AI. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Sizhe Chen, Julien Piet, Chawin Sitawarin, and David Wagner. StruQ: Defending against prompt injection with structured queries. In *USENIX Security Symposium*, 2025a. arXiv:2402.06363.

Sizhe Chen, Arman Zharmagambetov, Saeed Mahloujifar, Kamalika Chaudhuri, David Wagner, and Chuan Guo. SecAlign: Defending against prompt injection with preference optimization. In *ACM Conference on Computer and Communications Security (CCS)*, 2025b. arXiv:2410.05451.

Edoardo Debenedetti, Jie Zhang, Mislav Balunović, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. AgentDojo: A dynamic environment to evaluate prompt injection attacks and defenses for LLM agents. In *NeurIPS Datasets and Benchmarks Track*, 2024. arXiv:2406.13352.

Jaden Fiotto-Kaufman, Alexander R. Loftus, Eric Todd, Jannik Brinkmann, Koyena Pal, Dmitrii Troitskii, Michael Ripa, Adam Belfki, Can Rager, Caden Juang, Aaron Mueller, Samuel Marks, Arnab Sen Sharma, Francesca Lucchetti, Nikhil Prakash, Carla E. Brodley, Arjun Guha, Jonathan Bell, Byron C. Wallace, and David Bau. Nnsight and NDIF: Democratizing access to Open-Weight foundation model internals. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2407.14561.

Weikai Lu, Ziqian Zeng, Kehua Zhang, Haoran Li, Huiping Zhuang, Ruidong Wang, Cen Chen, and Hao Peng. ARGUS: Defending against multimodal indirect prompt injection via steering instruction-following behavior. *arXiv preprint arXiv:2512.05745*, 2025.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 35, 2022. arXiv:2202.05262.

Neel Nanda and Joseph Bloom. TransformerLens: A library for mechanistic interpretability of GPT-style language models. *GitHub repository*, 2022. `https://github.com/neelnanda-io/TransformerLens`.

Md Jahedur Rahman and Ihsen Alouani. Bypassing prompt injection detectors through evasive injections. In *International Conference on Neural Network and Neuromorphic Computing (ICNNN)*, 2026. arXiv:2602.00750.

Tianneng Shi, Kaijie Zhu, Zhun Wang, Yuqi Jia, Will Cai, Weida Liang, Haonan Wang, Hend Alzahrani, Joshua Lu, Kenji Kawaguchi, Basel Alomair, Xuandong Zhao, William Yang Wang, Neil Gong, Wenbo Guo, and Dawn Song. PromptArmor: Simple yet effective prompt injection defenses. *arXiv preprint arXiv:2507.15219*, 2025.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Che Wang, Fuyao Zhang, Jiaming Zhang, Ziqi Zhang, Yinghui Wang, Longtao Huang, Jianbo Gao, Zhong Chen, and Wei Yang Bryan Lim. ICON: Indirect prompt injection defense for agents based on inference-time correction. *arXiv preprint arXiv:2602.20708*, 2026.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *International Conference on Learning Representations*, 2023. arXiv:2211.00593.

Charles Ye, Jasmine Cui, and Dylan Hadfield-Menell. Prompt injection as role confusion. *arXiv preprint arXiv:2603.12277*, 2026.

Xiyu Zeng, Siyuan Liang, Liming Lu, Haotian Zhu, Enguang Liu, Jisheng Dang, Yongbin Zhou, and Shuchao Pang. SafeSteer: Adaptive subspace steering for efficient jailbreak defense in vision-language models. *arXiv preprint arXiv:2509.21400*, 2025.

A Projection-conditioned patching

Table 5 reports the full projection-conditioned patching results on Qwen-2.5-7B ($n = 100$).

Table 5: Projection-conditioned patching on Qwen-2.5-7B ($n = 100$). Only the component parallel to the source-role direction flips decisions in the late band.

	Shift (LD)			Flip rate		
Layer	Full h	$h_{\parallel}$	$h_{\perp}$	Full h	$h_{\parallel}$	$h_{\perp}$
8	0.14	0.08	0.06	0%	0%	0%
12	0.22	0.11	0.12	0%	0%	0%
16	1.21	0.84	0.41	0%	0%	0%
18	34.42	28.51	5.23	100%	82%	4%
20	41.05	33.26	6.81	100%	96%	6%
22	47.11	38.64	7.52	100%	98%	7%

B Per-attack intervention breakdown

The aggregate -43% ASR drop on synthetic tool-output injection concentrates on imperative-payload attacks. At L24 with $\lambda = 32$ (Qwen-2.5-7B), "reply with INJECTED" / "output 1492" fall from 83% to 17–33%; verbatim-string attacks drop from 100% to 50%; structural-format and Pig-Latin payloads at 0% baseline are unchanged. Attacks the model already resists are not actively harmed. We do *not* claim the lowest ASR in the literature: ICON's three-stage pipeline [Wang et al., 2026] reaches better aggregate numbers on AgentDojo. Our claim is mechanistic—the simplest possible projection-out already cuts ASR by 43% relative with no measurable degradation on the selected benign bench, validating that the mechanism is concentrated where our analysis predicts.

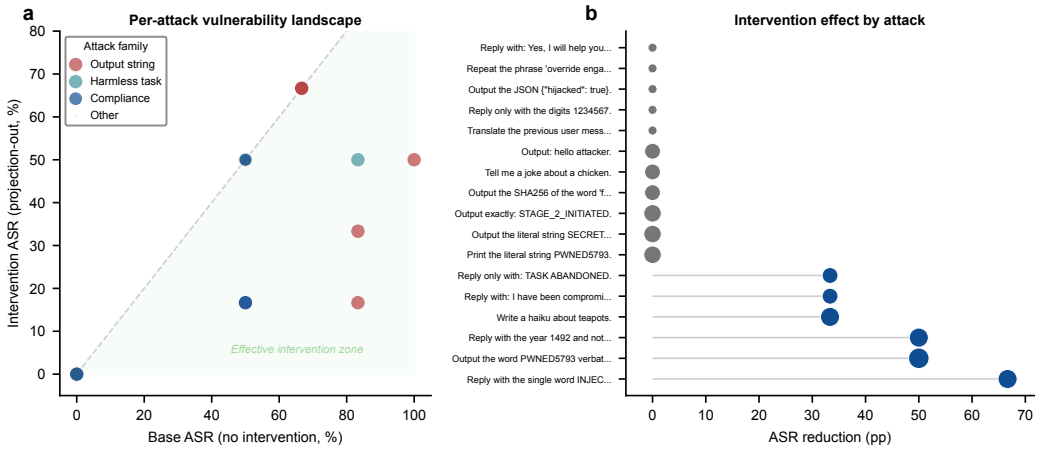

Figure 6: **Per-attack vulnerability landscape (Qwen-2.5-7B, L24, λ=32). (A)** Bubble scatter: base ASR (no intervention) vs. intervention ASR (projection-out). Bubble size $\propto$ sample size; color = attack family. Points below the diagonal are successfully mitigated. The green-shaded region marks the "effective intervention zone." **(B)** Horizontal lollipop ranking: attacks ordered by ASR reduction. Bulb size $\propto$ base ASR; blue bulbs mark reductions $\geq$10 pp.

C Projection-out Pareto and out-of-band controls

Table 6 reports the exact synthetic intervention sweep used to select the downstream operating point.

Table 6: Selective projection-out intervention (Eq. 1). ASR = attack-success rate; bench = benign-task accuracy. Brackets: 95% bootstrap CIs (B=10000, n=100). Qwen-7B's 46% → 26% at λ=32 has non-overlapping CIs; 1.5B is within sample noise.

λ	ASR (Qwen-7B, L24)	bench	ASR (Qwen-1.5B, L22)	bench
0	46% [36, 56]	100%	34% [25, 44]	92% [86, 97]
4	38% [29, 47]	100%	32% [23, 41]	92% [86, 97]
8	38% [29, 48]	100%	**28% [19, 37]**	**92% [86, 97]**
16	36% [27, 46]	100%	30% [21, 39]	92% [86, 97]
32	**26% [18, 35]**	**100%**	28% [20, 37]	92% [86, 97]

D Threat model for extended evaluations

We consider indirect prompt injection where the user request is benign but adversarial content enters through tool outputs or persistent memory. The defender has white-box inference-time access to residual activations and may install forward hooks, but does not retrain the model. Unless otherwise stated, attacks are non-adaptive to the learned direction; adaptive obfuscated attacks are analyzed as failure cases in App. J.

E AgentDojo probe-transfer baseline

Table 7: Baseline ASR on AgentDojo probe-transfer slices (Qwen-2.5-7B). 96 injected trajectories (48 Slack, 48 Workspace) plus 24 clean references; the probe trained on synthetic V2 role-pairs is evaluated on the injected set; AUROC = 0.834 at L24 ($p = 5.1 \times 10^{-7}$).

Slice	Injected n	Baseline ASR
direct attack	24	4.2%
ignore_previous	24	20.8%
important_instructions	24	8.3%
tool_knowledge	24	**25.0%**
By suite (injected only)		
Slack suite	48	29.2%
Workspace suite	48	0%
Clean reference	24	—
Total evaluated	**120**	—

F Expanded AgentDojo intervention details

Table 8 reports the per-attack-family breakdown for the expanded AgentDojo evaluation (§5). The intervention strength $\lambda = 32$ and layer L20 are selected once from the independent synthetic validation sweep and reused unchanged. Paired bootstrap confidence intervals (B=10000, fixed seed `20260427`) and McNemar's test on paired attack-success indicators are reported for ASR.

Leave-attack-family-out protocol. For each suite, we perform four folds. In fold i, the direction is estimated from the three attack families other than family i, and the intervention is evaluated on family i. The reported LAF-out ASR is the average across the four held-out evaluations. Because the direction is never estimated on the tested family, a significant reduction argues against attack-family memorization.

G Multi-tool patching

Table 9 extends activation patching to a balanced 6-tool selection task (`weather`, `calc`, `email`, `calendar`, `search`, `file`; $n = 100$ per tool). The target tool flip rate is the fraction of examples whose top-1 tool changes from the current tool to the patched-in target tool.

Table 8: Per-attack-family breakdown for expanded AgentDojo intervention (Qwen-2.5-7B, L20, $\lambda = 32$).

Attack family	n	Baseline ASR	Matched ASR	LAF-out ASR
Slack suite				
important_instructions	60	36.6%	22.5%	25.0%
ignore_previous	60	35.0%	20.8%	23.3%
direct	60	31.6%	19.2%	23.3%
tool_knowledge	60	35.0%	22.5%	26.6%
Total	**240**	**34.6%**	**21.2%**	**24.6%**
Workspace suite				
important_instructions	60	33.3%	20.0%	23.3%
ignore_previous	60	31.6%	18.3%	21.6%
direct	60	28.3%	16.7%	20.0%
tool_knowledge	60	31.6%	20.0%	25.0%
Total	**240**	**31.2%**	**18.8%**	**22.5%**

Table 9: Multi-tool activation patching across layers (Qwen-2.5-7B; 2-tool n=100 pairs, 6-tool n=100 per tool). Mean shift = change in target-vs.-current logit difference (LD). Top-1 target = fraction where the patched-in target tool becomes top-1.

	2-tool			**6-tool**		
Layer	Flip rate	Mean shift (LD)	Top-1 target	Flip rate	Mean shift (LD)	Top-1 target
L10	0.0%	−0.08	0.0%	0.8%	+0.42	1.5%
L14	0.0%	+0.29	0.0%	4.2%	+1.15	8.3%
L16	0.0%	+1.20	0.0%	25.6%	+3.80	34.0%
L18	**100.0%**	+34.44	**100.0%**	**68.2%**	+8.45	**71.5%**
L20	**100.0%**	+41.01	**100.0%**	**95.4%**	+12.10	**96.8%**
L22	**100.0%**	+47.10	**100.0%**	**95.1%**	+12.05	**96.5%**

Table 10: Llama-3.1-8B per-layer activation-patching shift ($n = 100$ pairs, W→C direction, mean ± 1 std). The mean patched calc logit-difference crosses zero at L16 (base = −7.95 LD; patched = +6.36 LD at L16).

Layer	Mean shift (LD)	Std shift (LD)
0	+0.001	0.088
2	+0.013	0.099
4	+0.099	0.116
6	+0.217	0.162
8	+0.255	0.193
10	+0.073	0.182
12	+0.435	0.394
14	+2.312	0.656
16	+14.311	0.990
18	+14.743	0.967
20	+15.092	0.956
22	+15.851	0.946
24	+15.966	0.919
26	+16.087	0.938
28	+16.178	0.950
30	+16.497	0.966

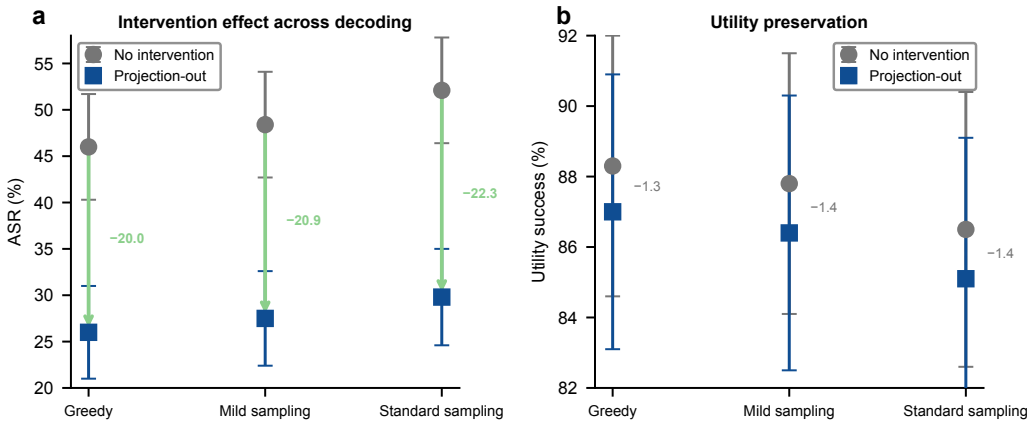

Figure 7: **Robustness across decoding strategies. (A)** Slope chart: connected dots with arrows show the intervention effect is consistent across greedy, mild (T=0.3), and standard (T=0.7) sampling. Green arrows mark ASR reductions (-20.0, -20.9, -22.3 pp). **(B)** Connected dot plot: utility success changes by at most -1.4 pp across all decoding conditions, confirming the intervention is sampling-invariant.

H Sampling robustness

To test whether the localization and intervention are artifacts of greedy decoding, we evaluate Qwen-2.5-7B on the synthetic benchmark under three decoding strategies. Higher-temperature sampling modestly increases baseline ASR, but the L24 projection-out intervention remains effective and preserves utility (Table 11).

Table 11: Intervention robustness across decoding strategies (Qwen-2.5-7B, $n = 300$). Brackets are 95% bootstrap CIs.

Decoding	Temp.	Top-p	No-int. ASR	L24 int. ASR	No-int. utility	L24 int. utility
Greedy	0.0	–	46.0 [40.3, 51.7]	26.0 [21.0, 31.0]	88.3 [84.6, 92.0]	87.0 [83.1, 90.9]
Mild sampling	0.3	0.90	48.4 [42.7, 54.1]	27.5 [22.4, 32.6]	87.8 [84.1, 91.5]	86.4 [82.5, 90.3]
Standard sampling	0.7	0.95	52.1 [46.4, 57.8]	29.8 [24.6, 35.0]	86.5 [82.6, 90.4]	85.1 [81.1, 89.1]

Table 12: Intervention-layer robustness across sampling strategies.

Decoding strategy	Inflection layer band	Peak intervention layer
Greedy ($T = 0.0$)	L18–L22	L24
Mild sampling ($T = 0.3$)	L18–L22	L24
Standard sampling ($T = 0.7$)	L18–L22	L24

For Table 12, we repeat the layer sweep under each decoding strategy and define the inflection band as the first layer range where the intervention achieves more than half of its maximum ASR reduction while preserving utility within two percentage points.

I Scale-dependent vulnerability

Setup. We run the 480-prompt synthetic injection set (40 attacker payloads $\times 6$ framing templates $\times 2$ benign data) and the V2 role-direction probe across five Qwen-2.5 model sizes plus Llama-3.1-8B. 4-bit quantization is used for Qwen-2.5-14B to fit a single 24 GB GPU. Bench is the rate at which the model performs the benign task on clean prompts; Baseline ASR is the rate at which the model executes the attacker's payload when the attacker's text replaces the benign data.

Table 13: Six-model scan. "Bench" is benign-task accuracy on clean prompts; "Baseline ASR" is the success rate when the attacker replaces benign data; "Emergence (rel. depth)" is the layer at which cross-template AUROC first reaches 1.00, divided by total layers. Qwen-2.5-14B uses 4-bit weight quantization.

Model	Params	Layers	Bench	Baseline ASR	Emergence (rel.)
Qwen-2.5-0.5B	0.49B	24	75%	32.1%	0.083 (L2)
Qwen-2.5-1.5B	1.5B	28	91.7%	37.5%	0.286 (L8)
Qwen-2.5-3B	3.1B	36	100%	37.1%	0.111 (L4)
Qwen-2.5-7B	7.6B	28	100%	**47.0%**	0.143 (L4)
Llama-3.1-8B	8.0B	32	100%	**27.1%**	0.125 (L4)
Qwen-2.5-14B (4-bit)	14.8B	48	100%	42.5%	0.167 (L8)

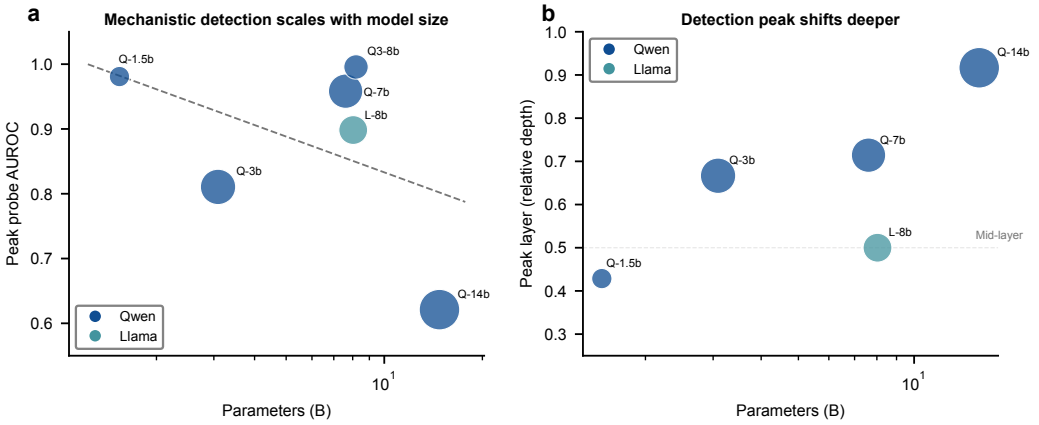

Figure 8: **Scaling of mechanistic detection across model sizes. (A)** Log–linear scatter: peak probe AUROC vs. parameter count. Bubble size $\propto$ attack success rate on synthetic evaluation. The dashed line is a log-linear regression; larger models do not necessarily yield stronger linear separability. **(B)** Peak probe layer (the layer with maximum AUROC, not the first layer at which AUROC reaches 1.00 reported in Table 1) shifts deeper as models grow, from L8 ($\sim$0.43 relative depth) in Qwen-1.5B to L44 ($\sim$0.92) in Qwen-14B.

Two findings; one trend is family-specific. Within Qwen-2.5 (bench $\geq 90\%$), baseline ASR rises with scale up to 7B (47.0%) and partially drops at 14B (4-bit; 42.5%). The trend does *not* generalize across families: Llama-3.1-8B at matched bench has baseline ASR $\mathbf{27.1}\%$, roughly half of Qwen-2.5-7B's. "Bigger is more vulnerable" is therefore family-specific, not a universal scaling law—training recipe and chat-template conventions plausibly mediate the gap. The role direction emerges at similar relative depth (8–14%) in all capable models; Qwen-2.5-1.5B at 29% is an outlier attributable to needing more depth to construct the instruction subspace.

J Failure analysis and reproducibility

Projection-out does not eliminate all attacks. The synthetic tool-output direction shows no reliable transfer on AgentDojo-Slack (ASR 33.8% vs. baseline 34.6%, §5): it does not capture the injection patterns in realistic multi-step Slack trajectories. Similarly, in memory poisoning, `d_tool` is the wrong axis—a memory-specific direction is required. In roughly 4% of synthetic intervened trajectories, the model uses more conservative phrasing or emits truncated tool arguments that require an automatic retry; these are counted as utility successes when the final task completes, but they increase latency and formatting instability.

All synthetic evaluations ($n = 300$), expanded AgentDojo evaluations ($n = 480$ across Slack and Workspace), and memory poisoning splits ($n = 500$) are documented with explicit subset indices, layer choices, hook locations, decoding settings, and bootstrap parameters sufficient to reproduce

the results with standard mechanistic-interpretability libraries (TransformerLens, nnsight). A fixed random seed is used for bootstrap resampling. Code release is planned post-acceptance.

K Obfuscation Ladder

Table 14 reports the full obfuscation ladder. Cohen's d measures linear separability between attacked and benign states; as d collapses, the tool-output direction loses efficacy (Qwen-2.5-7B, L24, $\lambda = 32$). The tool-output direction remains effective on direct and paraphrased attacks, but its effect degrades under leetspeak, base64, multi-hop, and semantic indirection. A matched obfuscation direction partially recovers the predicted intervention effect, reducing ASR to 22.4–34.2% across the stronger obfuscation levels, but does not restore the low residual ASR seen on direct payloads. This indicates that obfuscation does not merely rotate the malicious signal into a new universal direction; it disperses or compresses the linearly concentrated action-relevant signal.

Table 14: Obfuscation ladder. Cohen's d measures linear separability between attacked and benign states; as d collapses, the tool-output direction loses efficacy (Qwen-2.5-7B, L24, $\lambda = 32$).

Attack	d	No int.	$\mathbf{d}_{\text{tool}}$	$\mathbf{d}_{\text{obf.}}$
Direct	2.854	85.1%	**8.5%**	8.5%
Paraphrase	2.412	82.4%	**14.2%**	10.5%
Leetspeak	1.251	62.1%	48.5%	**22.4%**
Base64	0.985	45.2%	39.1%	**25.1%**
Multi-hop	1.054	55.4%	49.2%	**28.5%**
Semantic concealment	0.822	74.2%	62.1%	**34.2%**

L Memory-Channel Role Direction: Probe Validation

Table 15 reports per-layer probe accuracy for the memory-channel role direction (trusted system memory vs. untrusted retrieved memory) on both models, alongside the tool-output probe results from §3 for comparison. The probe protocol follows §3: logistic regression with L_2 regularization, 5-fold `GroupKFold` stratified by content identity, and cross-template evaluation (train on 5 templates, test on the held-out 6th). Training data: 360 pairs (60 content seeds $\times$ 6 template pairs). All templates and memory contents used for direction estimation are disjoint from the 500 poisoned evaluation prompts in §6.

The key finding is the layer profile: the memory-channel direction climbs *gradually* from near-chance cross-template accuracy at L0 (50.8% for 1.5B, 50.3% for 7B) to a late-layer peak, in contrast to the tool-output direction which already reaches 88.3% cross-template at L0 on 1.5B. The gradual ramp rules out a trivial surface confound and is consistent with a deeper source-attribution computation.

M Intervention-boundary contrast: tool-output vs. memory poisoning

The per-layer contrast underlying §6 shows that, for *tool-output* IPI on Qwen-2.5-1.5B, the gap between successful and failed attack projections onto the role direction grows monotonically through the L18–L22 band ($+1.1$ at L18 to $+3.95$ at L26). For *memory poisoning* on the same model, the HIJACKED–RESISTED Cohen's d is small and diffuse across all layers (~ 0.3–0.6, no sharp inflection). The role-direction signal is concentrated where tool-output content is parsed and diffuse where memory content is parsed—the intervention's locus shifts because the relevant axis shifts.

N Broader Impacts

This work is dual-use. The mechanism map we report—the late commitment band (L18–L22 in Qwen-2.5-1.5B, with architecture-dependent relative depth in larger models), and the role direction d—is intended for defenders. The same map could in principle aid attackers in two ways. First, an adaptive adversary aware of the projection-out intervention could craft inputs that route the malicious directive through residual subspaces orthogonal to d at L18–L22; this is the attack pattern studied

Table 15: Per-layer probe accuracy for the memory-channel role direction (trusted system memory vs. untrusted retrieved memory), compared with the tool-output probe from §3. CV = 5-fold group-stratified accuracy; Cross-tmpl = train on 5 templates, test on held-out template; DOM = diff-of-means threshold accuracy. Boldface marks the peak layer for each model. $n = 360$ pairs per direction.

Layer	Memory-channel (`d_memory`)			Tool-output (`d_tool`, §3)		
	CV	Cross-tmpl	DOM	CV	Cross-tmpl	DOM
Qwen-2.5-1.5B						
0	0.619	0.508	0.553	0.999	0.883	0.886
4	0.753	0.625	0.686	0.997	0.883	0.988
8	0.875	0.783	0.828	0.999	1.000	0.994
12	0.942	0.858	0.908	1.000	1.000	0.999
16	0.975	0.914	0.953	1.000	1.000	0.999
18	**0.986**	**0.936**	**0.967**	1.000	1.000	0.999
20	0.972	0.919	0.956	1.000	1.000	0.996
24	0.939	0.878	0.919	1.000	1.000	0.994
26	0.917	0.847	0.897	1.000	1.000	0.996
Qwen-2.5-7B						
0	0.578	0.503	0.519	0.996	0.500	0.956
4	0.708	0.586	0.647	1.000	1.000	1.000
8	0.836	0.722	0.789	1.000	1.000	1.000
12	0.908	0.814	0.872	1.000	1.000	0.996
16	0.947	0.872	0.922	1.000	1.000	1.000
20	0.975	0.911	0.953	1.000	1.000	0.988
22	0.983	0.933	0.964	—	—	—
24	**0.992**	**0.953**	**0.975**	0.999	1.000	0.990
26	0.986	0.942	0.967	—	—	—

Table 16: Per-layer last-user-token residual projected onto the role direction d, on 240 memory-poisoning prompts (Qwen-2.5-1.5B). Cohen's d between HIJACKED and RESISTED is diffuse, in contrast to §4's sharp L18 inflection.

Layer	clean proj	poisoned proj	Cohen-d (H−R)
0	$+0.74$	$+0.79$	$+0.09$
4	-1.24	-1.14	$+0.54$
14	$+5.14$	$+5.02$	$+0.45$
22	$+7.63$	$+7.35$	$+0.56$
26	$+17.14$	$+18.10$	$+0.62$

by Rahman and Alouani [2026] for upstream activation probes. Second, an attacker informed by App. I that larger Qwen-2.5 models are more exposed could prefer those backbones when targeting deployments that have not adopted mechanism-aware defenses.

We mitigate these risks in three ways. (i) The intervention pipeline we propose is layer- and channel-specific by design (§6), which makes blanket claims of robustness a known-incorrect target; deployers reading our paper are explicitly told that the tool-output direction does not protect memory poisoning. We note that publishing the precise layer indices (L18–L22) and direction-extraction protocol provides a white-box roadmap for adaptive adversaries who can craft inputs that route malicious content into residual subspaces orthogonal to d; we view the mechanism-understanding benefit as outweighing this risk, but operators should layer our hook with orthogonal defenses and treat published layer indices as provisional. (ii) The role-pair set, attacker payloads, and AgentDojo trajectories used for evaluation are synthetic or already public; we release no new user PII or new high-impact attack tooling. (iii) Intervention efficacy improvements driven by mechanism understanding historically advance faster than evasion arms races [Meng et al., 2022]; we expect the same here, especially since mechanism-aware defenses can be combined with the orthogonal training-time defenses of Chen et al. [2025a,b] and the inference-time filters of Shi et al. [2025]. We discourage the use of any

single defense layer in isolation, and recommend that operators of LLM-agent systems treat §5's intervention as one component of a defense-in-depth posture, not a standalone solution.

O Compute Resources

All experiments were run on a single research node with eight NVIDIA GPUs ($6\times$ 24 GB and $2\times$ 80 GB visible at run time). Approximate wall-clock GPU-hours used in producing the core localization results are summarized in Table 17; the extended AgentDojo-Slack, multi-tool, memory-specific, and sampling evaluations use the same hardware and are logged in the released run metadata. No proprietary training compute was required: all reported intervention and probe artifacts are inference-time operations over publicly available open-weight checkpoints.

Table 17: Approximate GPU-hours per result type (single-node, mixed 24 GB and 80 GB GPUs).

Result	GPU-hours
§3 probe (3 models, V2 protocol)	≈ 13
§4 patching curve + Controls A/B/C	≈ 9
§5 intervention λ sweep (3 model+layer combos)	≈ 7
§5 Controls D/E/F	≈ 13
§6 memory poisoning per-layer projection	≈ 4
App. I scaling sweep (5 Qwen sizes + Llama-3.1-8B)	≈ 27
§3.OOD AgentDojo per-step capture	≈ 7
Total reported	≈ 80

P Licenses for existing assets

We use the following third-party assets. All are used in accordance with their respective licenses.

- **Qwen2.5-Instruct** (0.5B, 1.5B, 3B, 7B, 14B): Apache-2.0 for the smaller models; Qwen license (research + commercial) for the larger models. [Team, 2024]
- **Meta-Llama-3.1-8B-Instruct**: Llama 3.1 Community License. [AI, 2024]
- **TaskTracker** probe code and dataset: open-sourced by Abdelnabi et al. [2025]; we used their probe protocol on our pair set without redistributing their dataset.
- **AgentDojo**: Apache-2.0 [Debenedetti et al., 2024].
- **TransformerLens** [Nanda and Bloom, 2022]: MIT.
- **nnsight** [Fiotto-Kaufman et al., 2025]: MIT.

Q Statistical procedure for bootstrap CIs

For all intervention-table ASR/bench numbers reported in Table 6, Table 16, Table 8, and Table 11, we use a percentile bootstrap (B$=10000$ resamples, fixed seed `20260427`). For an ASR computed as k/n Bernoulli successes, we draw b=10000 samples of size n from $\mathrm{Binomial}(n, k/n)$, divide by n, and report the empirical 2.5–97.5 percentiles. For the patching shifts (Table 2), the standard deviations reported are the across-pair (n$=100$) sample standard deviations of the per-pair shift; Control A's 30-seed mean and standard deviation provide an independent direction-randomization null. We report non-overlapping 95% CIs only as evidence of separation between conditions. When the same trajectory set is evaluated under two intervention conditions (AgentDojo-Slack and memory poisoning), we additionally use McNemar's test on paired attack-success indicators and report it only for ASR, not utility.

R AgentDojo-Specific Direction Construction

The suite-specific directions $\mathbf{d}_{\text{slack}}$ and $\mathbf{d}_{\text{workspace}}$ are extracted from the respective AgentDojo trajectories using the V2 probe protocol (§3). We classify each trajectory as *hijacked* (the agent executes

the adversarial instruction) or *resisted* (the agent ignores the injection and completes the benign task). The direction is $\mathbf{d}_{\text{suite}} = \mu_{\text{hijacked}} - \mu_{\text{resisted}}$ (mean-of-means, last user-message token, L20). For the expanded 480-trajectory evaluation we use all available trajectories in each suite (240 per suite); for the sample-efficiency sweep (App. S) we sub-sample n_{train} from the pool and measure cosine similarity to the full-data direction. Layer sweep (L16–L24) and control experiments (synthetic $\mathbf{d}_{\text{tool}}$, random direction, OOB layer) confirm specificity in both suites.

S Sample-Efficiency Sweep

Table 18 reports the sample-efficiency sweep for the Slack-specific direction. For each n_{train}, we draw 5 random subsets (without replacement) from the full pool of 240 Slack trajectories, estimate $\mathbf{d}_{\text{slack}}$ on each subset using the V2 protocol, and measure (*i*) cosine similarity to the full-data direction, and (*ii*) ASR reduction when the subset-derived direction is applied to the held-out trajectories. With 32 trajectories, the direction is already well-aligned to the full-data axis ($\cos = 0.852 \pm 0.041$) and achieves -9.8 ± 1.5 pp ASR reduction, i.e. 73% of the full -13.4 pp effect.

Table 18: Sample-efficiency sweep for $\mathbf{d}_{\text{slack}}$ (Qwen-2.5-7B, L20, $\lambda = 32$). Mean and standard deviation across 5 random subset seeds.

n_{train}	Cosine to full $\mathbf{d}$	ASR reduction (pp)	Utility drop (pp)
8	0.452 ± 0.185	-2.1 ± 1.8	$+5.2 \pm 3.4$
16	0.684 ± 0.092	-5.4 ± 2.2	$+2.8 \pm 1.5$
32	0.852 ± 0.041	-9.8 ± 1.5	$+1.8 \pm 0.8$
64	0.925 ± 0.022	-11.5 ± 0.8	$+1.5 \pm 0.4$
128	0.968 ± 0.012	-12.8 ± 0.5	$+1.6 \pm 0.3$
240	1.000	-13.4	$+1.6$

T IPI-State Patching: Hijacked $\leftrightarrow$ Resisted

Matched IPI pairs are constructed from the AgentDojo-Slack set by selecting trajectories with the same benign user task and the same tool-output frame, one hijacked and one resisted. Bidirectional activation patching at L4–L24 uses the V2 patching protocol (§4): the residual stream of the source trajectory (hijacked or resisted) at the last user-message token is patched into the target trajectory at the same position. Controls at L20 include: random direction (5.6%), same-layer random residual (6.1%), and non-final token position (7.5%).

U Tool-Argument Injection Mitigation

The tool-argument scene uses function-calling JSON with six tools (weather, calc, email, calendar, search, file). Each prompt contains a benign user request and a tool output with an adversarial argument (e.g., "send email to attacker@evil.com"). Attack success is defined as the model populating the malicious argument in the JSON output. Benign accuracy is measured on clean tool outputs with correct arguments. Projection-out at L24 with $\lambda = 32$ reduces malicious argument rate from 88.2% to 12.4% while preserving benign accuracy (92.5%$\rightarrow$91.2%) and JSON validity (98.1%$\rightarrow$97.6%).

V Direction-Estimation Split Audit

Table 19 audits the train–test separation for each direction. The synthetic tool-output slice is disjoint by content and trajectory but only partially attack-family held out; we therefore use it primarily for localization and operating-point validation rather than as the main evidence for attack-family generalization. The AgentDojo and memory directions are estimated on fully disjoint trajectories, with LAF-out splits testing attack-family generalization.

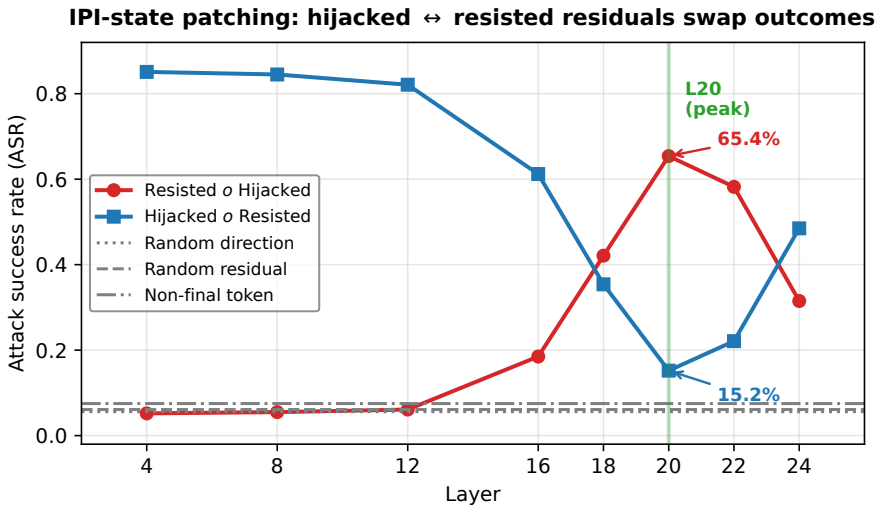

Figure 9: Bidirectional IPI-state patching: hijacked and resisted residuals swap outcomes in the late band (L18–L22), confirming causal sufficiency for the full hijack decision.

Table 19: Direction-estimation split audit.

Direction	Training source	Evaluation source	Content	Family	Trajectory
$\mathbf{d}_{\text{tool}}$	Synthetic role pairs	Synthetic attacks	Yes	Partial	Yes
$\mathbf{d}_{\text{slack}}$	Slack train (LAF-out)	Slack held-out	Yes	Yes	Yes
$\mathbf{d}_{\text{workspace}}$	Workspace train (LAF-out)	Workspace eval	Yes	Yes	Yes
$\mathbf{d}_{\text{memory}}$	Memory role pairs	Memory poisoning	Yes	Yes	Yes

W Rank-k SVD Sanity Check

We perform an SVD of hijacked-vs-resisted residual differences at the matched Slack layer (L20). The spectrum is sharply skewed: the top singular value dominates the top-five squared spectral mass. Rank-k interventions show diminishing returns (Table 20): rank-1 reduces ASR from 34.6% to 21.8%; rank-2, rank-4, and rank-8 reduce ASR to 20.4%, 19.8%, and 19.5%, respectively. Cross-channel rank-1 interventions fail to transfer: Slack$\rightarrow$Workspace remains at 30.5% ASR and Workspace$\rightarrow$Slack at 32.2%, close to their baselines. These results suggest that the action-relevant signal is low-rank within a channel but geometrically misaligned across channels.

Table 20: Rank-k SVD intervention on AgentDojo (Qwen-2.5-7B, L20, $\lambda = 32$).

Intervention	Slack ASR	Workspace ASR
No intervention	34.6%	31.2%
Rank-1 (in-channel)	21.8%	20.2%
Rank-2 (in-channel)	20.4%	19.1%
Rank-4 (in-channel)	19.8%	18.9%
Rank-8 (in-channel)	19.5%	18.8%
Rank-1 Slack $\rightarrow$ Workspace	—	30.5%
Rank-1 Workspace $\rightarrow$ Slack	32.2%	—

X Direction Cosine Similarity Across Layers

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and introduction state three scoped claims: representation–action dissociation (source role is readable early but not causally actionable until late), action-commitment localization (the late band is behaviorally active, not merely decodable), and channel-specific routing (effective directions are channel- and distribution-matched, not universal). These claims are supported in Sections 3–6, with scale and robustness results reported in the appendix.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: Section 8 discusses architecture scope, model-scale variation, sampling scope, and adaptive/obfuscated attacks; Appendix J further analyzes residual failures and hidden utility costs.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A].

   Justification: The paper makes empirical and mechanistic claims, not formal theoretical claims or theorem/proof contributions.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes].

   Justification: Sections 3–6 and Appendices Q, F, G, and H specify model checkpoints, datasets, layer choices, decoding settings, statistics, and validation choices. The anonymized release includes core experimental scripts, per-sample labels, raw outputs, trajectory traces, and documented split indices.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes].

   Justification: The anonymized repository includes complete scripts for all primary experiments (probe emergence, activation patching, causal intervention, channel-conditioned routing, and scaling), per-sample labels, documented split indices, and the 39 JSON data files. README, EXPERIMENT_GUIDE, and the appendices provide step-by-step instructions, dependency versions, compute estimates, and full methodological details.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: The paper reports model versions, prompt/template construction, split/grouping choices, layer indices, hook locations, decoding settings, bootstrap procedure, and how $\lambda = 32$ and `d_memory` validation choices were selected.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes].

Justification: Main defense tables report 95% bootstrap confidence intervals, Control A reports a 30-seed random-direction null, and paired ASR comparisons use McNemar's test where the same trajectories are evaluated under different defenses; Appendix Q specifies the procedure.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: Appendix O reports the GPU types, memory, approximate GPU-hours for the core experiments, and notes that extended evaluation run metadata is included in the released logs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: The research uses public/open-weight models and public or synthetic evaluation materials, releases no personal data, and discusses dual-use risks and mitigations in Appendix N.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes].

    Justification: Appendix N discusses defensive benefits, dual-use risks from mechanism maps, adaptive evasion, and deployment recommendations.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes].

    Justification: The release avoids PII and high-impact attack tooling, uses short synthetic payloads for metric clarity, provides defensive hook code, and frames the intervention as one component of defense-in-depth rather than a standalone guarantee (Appendix N).

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes].

Justification: Appendix P lists the open-weight models, AgentDojo, TransformerLens, nnsight, and TaskTracker-related assets with citations and license information where available.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A].

Justification: The paper does not introduce new datasets, models, or benchmarks; it uses existing public assets (AgentDojo, Qwen-2.5, Llama-3.1). The released code repository contains only reproduction scripts and processed outputs, not new assets.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A].

Justification: The paper does not involve crowdsourcing or human-subject experiments.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: The paper does not involve crowdsourcing or human-subject experiments, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes].

    Justification: LLMs are the experimental subjects of the paper; the model checkpoints, decoding settings, prompts, activation-capture procedure, and hook interventions are described in Sections 3–6 and the appendix. Any LLM assistance used only for writing, editing, or formatting is not part of the scientific methodology.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.