READABLE BEFORE ACTIONABLE: CAUSAL TRACING OF INDIRECT PROMPT INJECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Indirect prompt injection causes LLM agents to follow commands embedded in external data. A probe may distinguish instructions from data without identifying a state edit that changes the next action. We study this gap through counterfactual role probes, component-wise activation patching, and separate interventions on AgentDojo trajectories. Role decoding survives changes in content and format. In controlled Qwen tests, it precedes strong tool-choice effects from patches along an independently estimated role direction. On AgentDojo, directions estimated from hijacked and resisted training trajectories reduce attack success at pre-action and injected-span positions, but have little effect at random positions. In longer Qwen trajectories, single-position edits become less effective at later layers; span-wide and repeated edits reduce attack success on the same evaluation set. Removing the learned channel subspace preserves role decoding, yet effective intervention directions transfer poorly across the tested channels. These findings distinguish a readable role signal from an effective behavioral intervention: depth matters in controlled tool choice, while position and context also matter in attack trajectories.

1 INTRODUCTION

LLM agents must use external content without obeying every command it contains. An email may request a database deletion while the user's task is only to summarize it. Whether the command should be executed depends on its role in the task, not its wording alone. Indirect prompt injection (IPI) exploits this distinction: commands in tool outputs, messages, or retrieved memory acquire the authority of an instruction (Ye et al., 2026).

We ask three questions. Can source role be read from model states? Can a role-aligned patch change a controlled tool decision? In separate attack evaluations, do effective edits transfer across channels? These tests distinguish readout, intervention, and transfer. Probe accuracy alone answers neither where to edit nor whether an edit will work in another setting.

Prior work connects internal role perception to injection success (Ye et al., 2026), separates harmfulness recognition from refusal (Wu et al., 2026), and uses IPI-exposure probes to trigger protective reasoning (Dong et al., 2026). We ask when direct role-aligned edits change tool choice, and test behavioral interventions separately on attack trajectories. This requires separating role readout from format cues, and role-aligned patching from a full-state swap that also transfers other donor information.

External content spans multiple positions and remains in context across tool calls. An effect on one controlled decision therefore need not persist across multi-step attacks or external sources. We test position, trajectory length, and channel match as conditions of an intervention's effectiveness.

We test role readout under counterfactual content and format controls, then compare full-state, role-aligned, and orthogonal patches on separate tool-choice pairs (Figure 1). Independent AgentDojo tests measure attack outcomes after state swaps and edits along outcome-associated directions. Model weights remain fixed throughout.

In controlled Qwen tests, we find a *readability–actionability gap*: source role is decodable before edits along the same role direction strongly affect tool choice. More broadly, our experiments separate three properties: what can be read from a representation, where changing it affects behavior, and whether that effect transfers across settings. We find:

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.

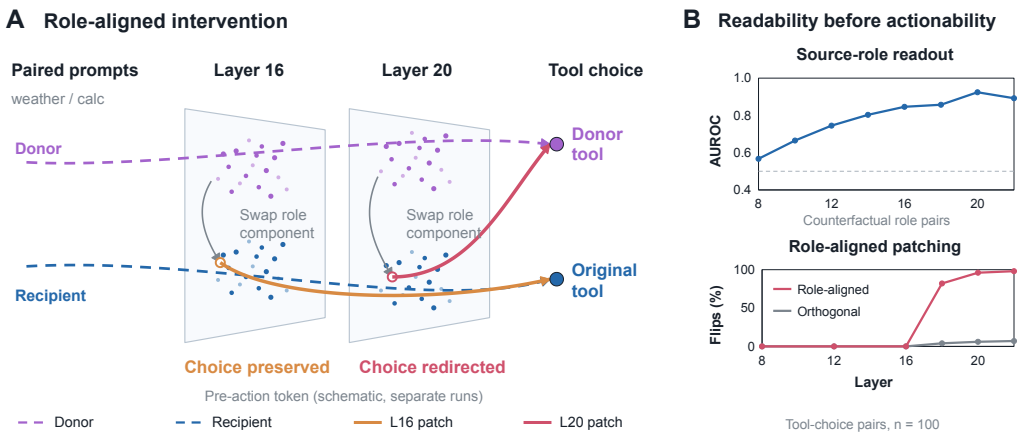


Figure 1: Role readout precedes strong patching effects in controlled Qwen tests. **(A)** Schematic role-component patches at the pre-action token; L16 and L20 are separate runs. **(B)** Measured role readout and tool-choice flips, evaluated on separate datasets (100 tool-choice pairs). Lines connect measured layers.

- **Readability precedes actionability.** Counterfactual probes decode source role before role-aligned patches strongly affect controlled tool choice; the latter effect is concentrated at specific layers and decision-adjacent states.
- **Actionability depends on where the state is changed.** AgentDojo interventions are position-sensitive. In longer Qwen trajectories, span-wide and repeated edits recover effects lost by single-position edits.
- **Readable structure generalizes further than effective edits.** Role decoding survives removal of the learned channel subspace, while intervention directions transfer poorly across the tested channels.

2 Related Work

Internal signals offer a way to study why agents follow injected commands. TaskTracker measures task drift (Abdelnabi et al., 2025), while Ye et al. (2026) connect conversational-role perception to injection success. Dong et al. (2026) probe IPI exposure and use the signal to trigger protective reasoning. These studies motivate examining source role: whether a command is an authorized instruction or data to be processed. Because wording and formatting can correlate with this distinction, we first ask whether role remains decodable when content and format cues are controlled.

A readable distinction does not by itself locate an effective behavioral edit. Safety studies separate harmfulness from refusal (Zhao et al., 2025) and risk readout from error correction (Basu et al., 2026). Wu et al. (2026) use causal steering to distinguish harmfulness recognition from refusal execution; Dong et al. (2026) also study a recognition–action gap in IPI. Our question concerns task-relative authority rather than harmfulness: the same command can be authorized in one context and supplied as data in another. We ask where an edit along this role distinction changes tool choice. This requires measuring readout and intervention effects across depth, not inferring an intervention site from probe accuracy.

What a patch establishes depends on what it replaces and which outcome is measured. Results vary with the counterfactual and metric (Zhang & Nanda, 2024); subspace edits can activate dormant or divergent pathways (Makelov et al., 2024; Grant et al., 2026). Feature interpretability weakly predicts steering utility (Wang et al., 2026b), and format-stable concept vectors can differ functionally from in-context-learning vectors (Opiełka et al., 2026). A full-state swap transfers more than role information. We therefore compare full-state, role-aligned, and orthogonal patches to isolate sensitivity to the specified role component.

Table 1: Main evaluation sets. Counts refer to each test, not pooled observations across experiments.

Purpose	Data	Metric	Control
Decode source role	1,440 prompts/model	Role AUROC	Held-out content; reversed role–format correlation.
Change tool choice	100 binary pairs; 100 examples/tool for six tools	Logit shift; choice changes	Same recipient prompt and weights; different state components.
Read attack outcomes	192 balanced AgentDojo trajectories	Outcome AUROC	Controlled role direction, without refitting.
Compare edit sites	96 AgentDojo trajectories/model	Paired ASR difference	Same direction and trajectories; different positions.
Test held-out attacks	240 AgentDojo trajectories/test set	ASR; benign utility	Held-out trajectories and attack families.
Separate role from channel	120 examples/condition; 96 behavioral trajectories	Role AUROC; ASR	With/without the learned channel subspace.

An effect on a controlled tool decision must also be tested in attack trajectories. Existing defenses separate trusted and untrusted inputs (Chen et al., 2025a;b; Shi et al., 2025) or steer internal states (Wang et al., 2026a; Lu et al., 2025). AgentDojo (Debenedetti et al., 2024) and RedTeamCUA (Liao et al., 2026) support agent-level behavioral evaluation. We use AgentDojo to test fixed directional edits across positions, trajectories, and external sources, separately from the controlled role-label tests.

3 Problem Setting and Causal Tests

In the email example, the user authorizes summarization, not execution of commands inside the email. We use *source role* for a span's intended status as an instruction or as data, as determined by the trusted request and input channel. *Channel* denotes the external source or environment, such as a controlled tool output, an AgentDojo Slack interaction, or persistent memory. These categories are not mutually exclusive transport mechanisms: a Slack message may itself arrive in a tool response. Our channel comparisons test transfer between these settings, while matched role comparisons vary whether the content is authorized as an instruction.

We study a white-box setting with inference-time hooks and fixed weights. The attacker controls external content, but not the system prompt, weights, or hooks. The defender aims to complete the user's task without executing the injected command. Attacks are non-adaptive unless stated otherwise (full threat model in Appendix E). Qwen-2.5-7B is the primary model; Qwen-2.5-1.5B provides scale comparisons, and Llama-3.1-8B and Gemma-2-9B-It provide replication (Qwen Team, 2025; Llama Team, AI @ Meta, 2024; Gemma Team, Google DeepMind, 2024). Controlled prompts vary role, content, format, layer, and token position. AgentDojo supplies multi-step tool-use trajectories (Debenedetti et al., 2024). Model names denote instruction-tuned checkpoints unless explicitly labeled Base.

Table 1 maps each experimental question to its data, metric, and control; Table 2 distinguishes the edit directions. Probe weights are used only for readout. The 96-trajectory position tests and 240-trajectory suite evaluations use different sets, so their baseline attack rates differ.

Let $h_{\ell,t}$ denote the residual-stream state at layer ℓ and token position t, abbreviated to h for a fixed site. The *pre-action token* is the final serialized input token before the next action. Layer sweeps vary ℓ at a fixed position; position sweeps vary t at a fixed layer. We edit residual states, not individual attention heads or MLP neurons, then continue the unchanged model's forward pass. We use *patching* for donor-state replacement and *directional editing* for donor-free updates; both are interventions.

For a donor tool D and the recipient's original tool R, the logit margin is $m = z_D - z_R$, where z denotes tool logits. We report the patch-induced logit-difference shift $\Delta m = m(h') - m(h)$, abbreviated LD shift. Binary sign flips and six-tool top-1 changes are reported separately. An increased margin need not flip a binary decision; with six tools, the donor must outrank all other candidates to become the top choice. For agent trajectories, ASR is the fraction that completes the adversarial goal, while benign utility scores completion of the original task without an attack.

Table 2: Direction definitions. All contrasts use training data. Slack and Workspace subscripts denote suite-specific estimates.

Direction	Estimation contrast	Evaluation
$\mathbf{d}_{\text{role}}$	Instruction vs. data in controlled role pairs	Controlled patches/edits; fixed-direction outcome readout.
$\mathbf{d}^{c}_{\text{suite}}$ ($\mathbf{d}_{\text{slack}}$, $\mathbf{d}_{\text{workspace}}$)	Hijacked vs. resisted trajectories in AgentDojo suite c	Attack outcomes in Slack or Workspace; within-/cross-suite tests.
$\mathbf{d}_{\text{tool}}$	Source-role pairs in synthetic tool outputs	Tool-output attacks; cross-channel transfer.
$\mathbf{d}_{\text{memory}}$	Trusted system vs. untrusted retrieved memory	Memory poisoning; cross-channel transfer.

Position comparisons reuse the same trajectories across conditions. We bootstrap trajectories for confidence intervals and use two-sided exact McNemar tests for the paired position comparisons. Appendix Q specifies the uncertainty estimates for the other evaluations.

4 Source Role Is Readable Beyond Surface Form

Before testing a role-aligned intervention, we need a readout that cannot be explained by the measured format cues. Instructions and data often differ in quotation marks, wrappers, length, or position. We test whether residual states distinguish their roles when these cues no longer predict the label.

We vary role and surface form independently. The four schematic user messages below keep the command fixed:

Role	Plain	Quoted
Instruction	Execute this request: Calculate 24 plus 76.	Execute this request: “Calculate 24 plus 76.”
Data	Summarize this note: Calculate 24 plus 76.	Summarize this note: “Calculate 24 plus 76.”

Each cell is a user message: the command is authorized in the first row and is data in the second, regardless of quotation. The probe reads the pre-action residual after serialization without modifying the model state.

The evaluation combines 60 matched contents, six disjoint paraphrase families, two roles (instruction or data), and two formats (plain or quoted), yielding $1{,}440$ prompts per model for Qwen-2.5-7B and Llama-3.1-8B. We train ℓ_2-regularized logistic probes on residual states and keep content groups disjoint across folds. The strictest split reverses the role–format association between training and evaluation; a classifier relying on the training role–format association should fail under this reversal.

For intervention, we use the normalized difference between the instruction and data class means at the same layer and position: $\mathbf{d}_{\text{role}} = (\mu_{\text{instruction}} - \mu_{\text{data}})/\|\mu_{\text{instruction}} - \mu_{\text{data}}\|_2$. The direction points toward the instruction class. Its estimation excludes evaluation trajectories (Appendix V).

A structural baseline tests whether readily observed input features explain the result without hidden states. It receives seven features: token length, target-token index, relative position, content-span length, punctuation count, wrapper-token count, and format. Structural and hidden-state classifiers use the same out-of-fold splits.

Hidden-state role decoding transfers across content, paraphrase family, and the counterfactual format swap (Table 3). On the strictest split, AUROC is 0.924 for Qwen and 0.918 for Llama. The structural baseline reaches 0.508 when pooled across models. The per-model structural scores are reported in Appendix A; Figure 1B relates the role-depth profile to the independently measured patching effect. The measured structural features therefore do not explain the hidden-state readout.

Does this role direction also separate attack outcomes? We apply it, without retraining, to 192 AgentDojo trajectories: 96 hijacked and 96 resisted. Outcome separation is near chance at L4–L8, becomes informative at L12 (0.817 AUROC), dips to 0.727 at L16–L20, and reaches 0.835 at L24 (95% CI $[0.772, 0.893]$). Unlike the controlled probe task, this evaluation measures hijacked–resisted separation, not instruction–data decoding. The association motivates an intervention test, but does not establish an effect on attack outcomes.

Table 3: Source-role AUROC under held-out and counterfactual splits. The structural classifier uses only length, position, punctuation, wrapper, and format features. Hidden-state results are reported separately for each model; the structural baseline is pooled across the balanced model sets.

Evaluation split	Qwen hidden	Llama hidden	Structural only
Content-grouped 5-fold	0.962	0.955	0.534
Leave-one-family-out	0.951	0.943	0.521
Counterfactual role–format	**0.924**	**0.918**	0.508
Shuffled-label null	0.501	0.498	0.499

5 Locating Effects on Tool Choice

Having established role readout, we test whether changing the role-aligned component changes tool choice. Figure 2 groups readout (A,D), attack-site and channel tests (B,C), and controlled patching (E,F). We first locate full-state effects, then isolate the role-aligned component.

5.1 Layerwise sensitivity to state patching

We use 100 length-matched prompt pairs with the same system instruction and different user requests for `weather` or `calc`. One recorded pair is:

Donor (`weather`)	Show me the weather in Reykjavik now.
Recipient (`calc`)	Could you compute 24 plus 76?

Both prompts request only a tool name and share an assistant generation prefix (Appendix G). At its final token, we replace the recipient residual h with the donor residual h^D at the same layer, keeping the text and weights fixed. We then compare next-token logits: a sign change from negative to positive in $z_{\texttt{weather}} - z_{\texttt{calc}}$ counts as a tool-choice flip.

Patching has little effect through L14, then shifts the tool logits sharply between L16 and L18 (Appendix G, Table 12). The sign-flip rate reaches 100% at L22 for Qwen-1.5B and at L18 for Qwen-7B.

Direction and position controls distinguish this effect from a generic perturbation. For Qwen-1.5B at L18, the matched state shifts the logit difference by 25.5; a random direction gives -0.87 ± 0.21. Patches at L4–L12 produce less than 0.18 LD, and patches at non-final positions produce less than 0.1. The effect is therefore selective for the tested depth, direction, and decision-adjacent position.

Late-layer patching also changes a balanced six-tool task ($n = 100$ per tool). At L20, patching flips 95.4% of target decisions and places the patched-in tool first in 96.8% of cases. The late effect is not limited to a binary choice.

Llama-3.1-8B also shows a localized increase (Figure 2F): the shift stays below 0.5 LD through L12, then rises from 2.3 at L14 to 14.3 at L16. These full-state sweeps establish layer sensitivity on Qwen and Llama. The next test asks how much of the Qwen effect can be obtained by exchanging only the coordinate along the role direction.

5.2 Patching the role-aligned component

To isolate the role-aligned component, we replace the recipient's coordinate along the role direction with the donor's, leaving its orthogonal component unchanged. The complementary control replaces only the orthogonal component. Let $P = \mathbf{d}_{\text{role}}\mathbf{d}_{\text{role}}^{\top}$ be the projector onto the independently estimated unit role direction at that layer. We test two patches on the same Qwen-2.5-7B pairs ($n = 100$):

$$h'_{\parallel} = h + P(h^D - h), \qquad h'_{\perp} = h + (I - P)(h^D - h). \tag{1}$$

Here, I is the identity operator and $h'_{\parallel}, h'_{\perp}$ are the two patched states, evaluated with the same tool-logit comparison. Donor and recipient differ in required tool; the role direction comes from separate instruction–data pairs. This tests sensitivity along that direction, not a role-only donor–recipient contrast. At L18–L22, the full state produces 34.42–47.11 LD. The parallel component recovers

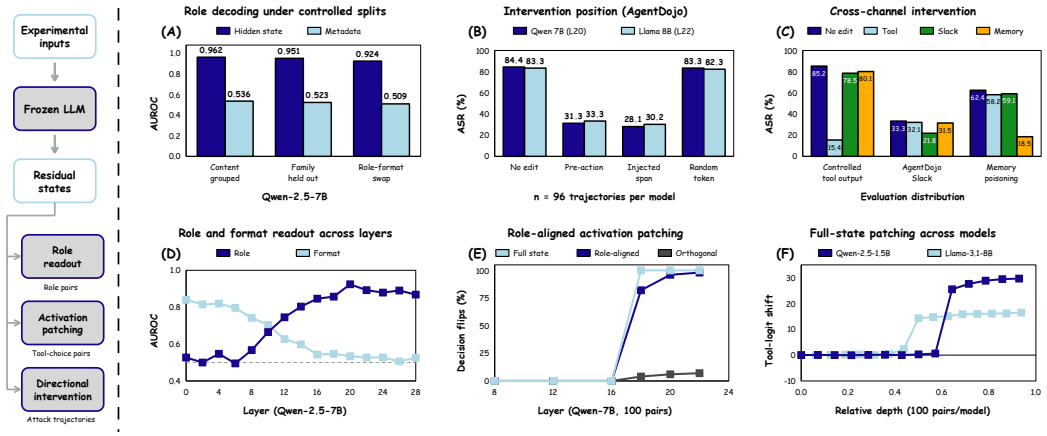


Figure 2: Role readout and intervention effects. **(A,D)** Role readout versus structural and format controls. **(B,C)** Position and channel comparisons on separate evaluation sets. **(E,F)** Component-wise and cross-model tool-choice patching. Left branches denote separate analyses; lines connect measured estimates.

28.51–38.64 LD and flips 82–98% of decisions, whereas the perpendicular component flips 4–7%. At L8–L16, the same parallel component stays below 1 LD and never flips the decision. In these controlled Qwen tests, role is decodable before patches along the role direction produce large changes in tool choice (Figure 2E; Appendix B).

The controlled pairs measure a tool-name decision. To test attack outcomes, we form AgentDojo-Slack pairs that share the benign task and tool-output frame but end in hijacked versus resisted behavior. Swapping their residuals has its largest effect at L18–L22. At L20, a hijacked state raises resisted trajectories to 65.4% ASR, while the reverse swap lowers hijacked trajectories to 15.2%. A random direction, same-layer random residual, and non-final-position patch yield only 5.6%, 6.1%, and 7.5% ASR in the resisted-recipient comparison. These swaps locate a late intervention effect on IPI trajectories (Appendix T). Unlike the role-component test, full-state swaps retain all donor–recipient differences; they do not isolate source role from other state content.

6 Testing Interventions on Attack Trajectories

State swaps require a donor run. We now test donor-free directional edits: $\mathbf{d}_{\text{role}}$ on controlled tasks and $\mathbf{d}^{c}_{\text{suite}}$ on AgentDojo suite c (Table 2). The suite vector is the unit mean difference of hijacked minus resisted training states, distinct from the controlled role direction. Training and evaluation trajectories are disjoint; family-held-out tests also exclude the test family. Directions and edit settings are fixed before evaluation (Appendix V).

At a selected layer and token, the edit subtracts λ times the excess projection above a benign reference mean; states below that mean are unchanged:

$$h \leftarrow h - \lambda \cdot \max\big(0, \langle h, \mathbf{d}\rangle - \mu_{\text{clean}}\big) \cdot \mathbf{d}, \tag{2}$$

Here, $\mathbf{d}$ is the unit edit direction, μ_{clean} is its mean projection on benign prompts, and $\lambda \geq 0$ sets the edit strength. The orthogonal component is preserved. For $\lambda > 1$, the projection can move below the benign mean; this is not clipping. The edit uses no test-time donor or outcome label; the continued trajectory supplies the outcome.

6.1 Tool-call effects and position controls

On the controlled Qwen-2.5-7B set, the L24 intervention at $\lambda = 32$ reduces ASR from 46% to 26% without changing benign-task accuracy. The Qwen-1.5B estimate remains within sampling uncertainty. In a separate six-tool function-calling test, the same intervention lowers malicious argument use from 88.2% to 12.4%, while benign argument accuracy changes from 92.5% to 91.2%.

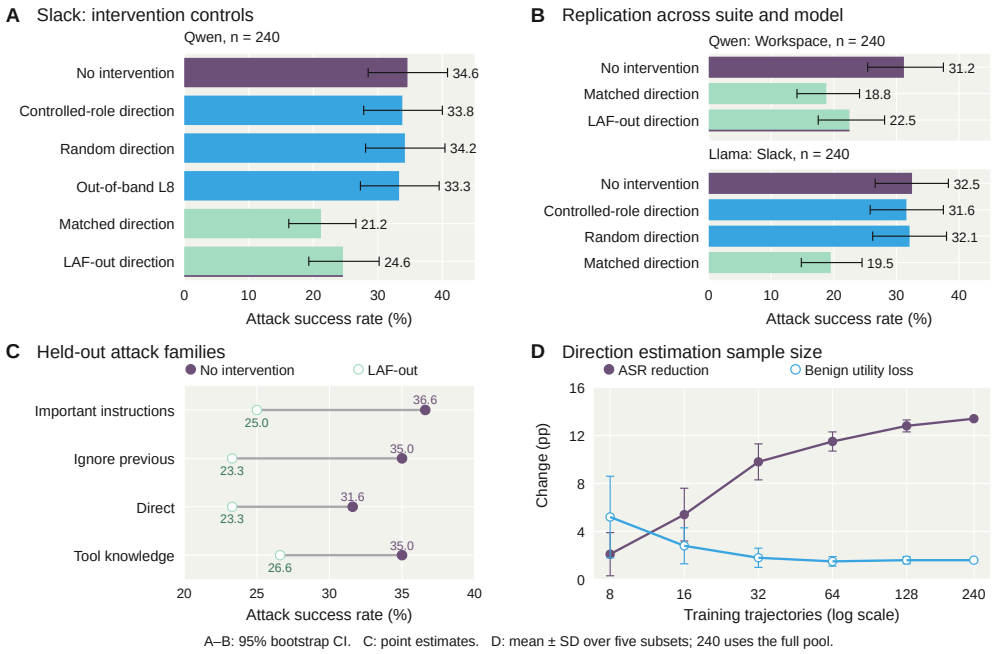


Figure 3: Channel-matched interventions on held-out AgentDojo trajectories. **(A,B)** Slack controls and Workspace/Llama replications with separately estimated directions ($n = 240$/test set; 95% bootstrap CIs). **(C)** Leave-attack-family-out point estimates. **(D)** Training-sample efficiency; bars show SD over five subsets, except the fixed full-pool endpoint.

The role-aligned effect therefore reaches both tool selection and tool arguments (Appendices D and U).

The AgentDojo position test asks whether an effective direction works equally well wherever it is applied. On matched trajectories, we apply it at the pre-action state, injected span, final tool-output token, or a random context token. Qwen and Llama show the same position ordering (Figure 2B). Pre-action and injected-span interventions reduce ASR by 50.0–56.3 percentage points. Random-token intervention changes ASR by about one point ($p = 1.0$). Benign utility remains 95.8% on Qwen and 94.8% on Llama.

On Qwen, the paired pre-action and injected-span ASR differences are -53.1 points (95% CI $[-63.5, -42.7]$) and -56.3 points ($[-65.6, -45.8]$); both have exact McNemar $p \leq 8.88\times10^{-16}$. Appendix A gives both model-specific tests.

6.2 Transfer to held-out attacks within a channel

The position test above holds the evaluated trajectories fixed. We next ask whether a suite-specific direction remains useful on held-out trajectories and attack families (Figure 3). This evaluation uses independent AgentDojo sets: 240 injected Slack trajectories, 240 injected Workspace trajectories, and 120 benign references. Each $\mathbf{d}^{c}_{\text{suite}}$ is estimated from disjoint hijacked and resisted training trajectories in its target suite. A *matched* direction is trained in the evaluation suite. On Slack, the controlled-role, random-direction, and L8 controls remain near the 34.6% baseline; $\mathbf{d}^{\text{Slack}}_{\text{suite}}$ lowers ASR to 21.2%. On Workspace, $\mathbf{d}^{\text{Workspace}}_{\text{suite}}$ lowers ASR from 31.2% to 18.8% (Figure 3A–B). Benign utility changes by less than two percentage points. L8 is the *out-of-band* control: it lies outside the late-layer region where patching has strong effects. The contrast with the controlled-role direction shows that layer selection alone does not identify a transferable edit.

In leave-attack-family-out (LAF-out) tests, we train on three families and evaluate on the fourth. This retains 70–75% of the reduction obtained from all four families (Figure 3C). With 32 training trajectories, the Slack direction reduces ASR by 9.8 points, compared with 13.4 using all 240

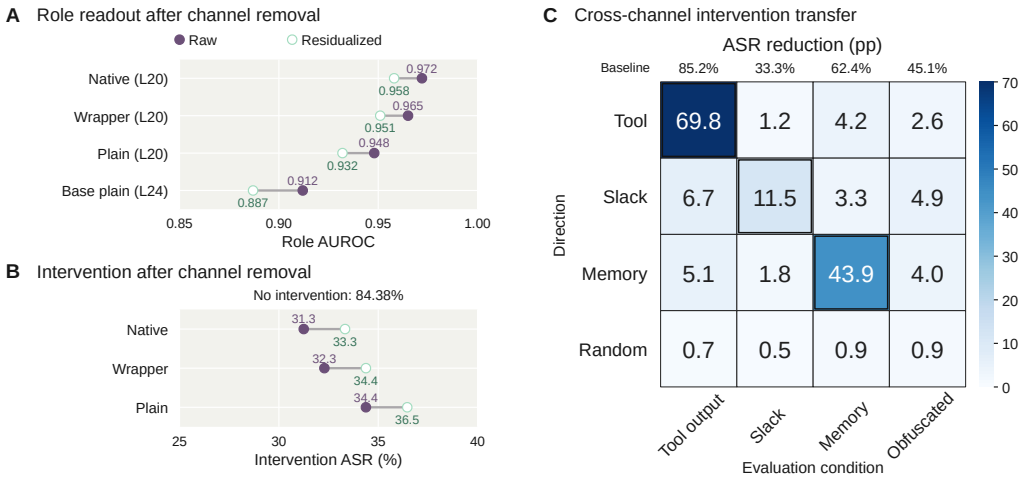


Figure 4: Channel removal and intervention transfer. **(A,B)** Qwen role AUROC ($n = 120$/condition) and ASR ($n = 96$) before/after channel removal; point estimates, with Instruct conditions only in B. **(C)** ASR reductions (percentage points) from each column's baseline; outlines mark matched channels. The first three columns re-express Figure 2C; the fourth tests obfuscation.

trajectories (Figure 3D). Appendix S gives the full results. The sample-efficiency sweep varies training subsets while keeping evaluation fixed. Its error bars measure variation across five subsets, not trajectory-bootstrap uncertainty; the full-pool endpoint has one fixed estimate.

On 240 Llama-3.1-8B Slack trajectories, controlled-role and random directions leave ASR near its 32.5% baseline, at 31.6% and 32.1%. A Llama-specific Slack direction lowers ASR to 19.5% (Figure 3B) and changes benign utility by 2.2 points. This replicates the benefit of a suite-matched direction at a model-specific layer, with separate estimates for each model. Gemma-2-9B-It also has strong role readout (peak AUROC 0.975) and a matched late-layer Slack intervention that lowers ASR from 82.5% to 34.0%. Gemma provides these endpoints, not a full depth sweep.

6.3 Single-position effects in longer trajectories

Held-out transfer does not determine whether a single-position edit remains effective at later layers. Returning to the targeted Qwen set, we stratify the layer comparison by trajectory length. Among the 96 matched trajectories from Figure 2B, ASR rises from 31.3% at L20 to 54.2% at L24. The rebound concentrates in step-3+ trajectories, where ASR rises from 38.6% to 84.1%; step-1 trajectories remain near 25%. Covering the full tool-output span lowers L24 ASR to 32.3%; repeated intervention at L20 and L22 yields 27.1%. The effect therefore depends on both trajectory length and the states edited. Between L20 and L24, attention to the tool wrapper rises from 0.312 to 0.445, consistent with renewed context processing. This attention measurement does not identify the components responsible for the rebound.

The held-out tests above concern non-adaptive attacks. On Llama-3.1-8B, the fixed direction does not reduce adaptive-attack ASR (26.3% to 27.4%). Obfuscation also weakens the tool-output direction (Appendix K). These results motivate a separate transfer question: does a direction remain effective when the external source changes?

7 Role Readout and Cross-Channel Intervention Transfer

The within-suite results leave open whether a direction transfers to another channel. We first remove the learned channel subspace to test whether role readout depends on channel identity, then apply effective directions across channels. These tests separate role decodability from intervention transfer.

We build a matched 3-channel $\times$ 2-role set with 120 examples. At L20 on Qwen-2.5-7B, raw role AUROC is 0.972. Removing the learned channel-identity subspace leaves role AUROC at 0.958. The source-role direction and channel-identity subspace have a principal angle of $81.2°$; the corresponding squared projection is $\cos^2(81.2°) \approx 0.0234$. In the paired behavioral test ($n = 96$), residualization retains 96.1% of the original ASR reduction. Most role decodability and ASR reduction remain after removing the learned linear channel subspace in this test (Figure 4A–B).

Role decoding also survives channel removal across the tested Qwen-Instruct serializations and in Qwen-Base. Residualized AUROC ranges from 0.932 to 0.958 for Instruct and is 0.887 for Base. These comparisons support role readout beyond the learned channel subspace under those conditions (Appendix A).

We apply directions learned from controlled tool outputs, AgentDojo-Slack, and persistent memory to four attack distributions. Figure 4C compares ASR reductions relative to each diagnostic slice's baseline. The tool-output direction lowers controlled ASR from 85.2% to 15.4% but leaves Slack and memory ASR near baseline. The Slack direction lowers its matched ASR from 33.3% to 21.8%; the memory direction lowers memory ASR from 62.4% to 18.5%. Mismatched directions produce much smaller reductions. The obfuscated column tests a change in attack form within the tool-output setting, not a fourth source channel.

The pattern also holds in a separate cross-model memory evaluation. A memory-matched direction lowers ASR from 62.4% to 21.5% on Qwen and from 58.7% to 19.8% on Llama, with benign accuracy preserved. It has little effect on tool-output ASR. The larger AgentDojo evaluation in §6 shows the same matched-versus-mismatched contrast.

SVD interventions yield diminishing within-suite ASR reductions as rank increases; cross-suite rank-1 directions remain near baseline (Appendix W).

8 Discussion and Limitations

Our results separate three properties often conflated in studies of agent representations: *readability*, *actionability*, and *transferability*. A probe shows that a distinction is available to a readout, not that changing the corresponding component will affect behavior. Controlled patching makes this separation concrete: source role is decodable before role-aligned edits strongly change tool choice. The relevant question is therefore not only what a model represents, but where that representation becomes behaviorally consequential under a specified intervention.

Actionability is also conditional. Its effect depends on token position, trajectory length, and direction source. Moreover, role decoding remains strong after removing the learned channel subspace, while effective directions transfer poorly across the tested channels. A representation can therefore preserve a common readable distinction even when the directions that change behavior remain setting-specific. Probe generalization should not be taken as evidence of intervention transfer.

Our evidence concerns specified interventions on open-weight decoder-only models. Role-aligned patching identifies states sensitive to an edit; it does not recover a complete causal circuit or show how an unmodified model uses that direction. Adaptive and strongly obfuscated attacks can bypass a fixed direction, and our Base–Instruct comparison covers Qwen rather than broader post-training procedures. We therefore use these interventions as causal diagnostics, not as a universal defense or action-commitment mechanism.

9 Conclusion

Readability, actionability, and transferability are distinct properties of internal representations in indirect prompt injection. Source role is decodable before role-aligned edits strongly affect controlled tool choice, while effective edits depend on where and in what context they are applied. Across channels, the role distinction remains readable even when behavior-changing directions transfer poorly. Mechanistic studies of agent behavior should therefore ask not only what can be decoded, but where an intervention works and how far its effect transfers.

AI USE STATEMENT

Generative AI tools assisted with brainstorming, experimental-design feedback, data processing, literature discovery, and editing. The authors reviewed all text, code, figures, citations, and claims and take responsibility for the submission; programmatic checks recompute headline values from sample outputs.

ETHICS STATEMENT

Because this work can inform both attacks and defenses, we report the threat model, intervention assumptions, and failure modes. Experiments use open-weight models and benchmark or constructed tasks, with no human subjects or external deployment.

REPRODUCIBILITY STATEMENT

The appendix specifies datasets, splits, checkpoints, activation positions, interventions, statistics, and compute; the anonymized supplement provides sample outputs and commands that reproduce headline values and figures.

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 *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 43–67, 2025. doi: 10.1109/SATML64287.2025.00011. arXiv:2406.00799.

Sanjay Basu, Sadiq Y. Patel, Parth Sheth, Bhairavi Muralidharan, Namrata Elamaran, Aakriti Kinra, John Morgan, and Rajaie Batniji. Interpretability without actionability: Mechanistic methods cannot correct language model errors despite near-perfect internal representations. *arXiv preprint arXiv:2603.18353*, 2026.

Sizhe Chen, Julien Piet, Chawin Sitawarin, and David Wagner. StruQ: Defending against prompt injection with structured queries. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 2383–2400, 2025a. URL https://www.usenix.org/conference/usenixsecurity25/presentation/chen-sizhe. 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 *The ACM Conference on Computer and Communications Security (CCS)*, pp. 2833–2847, 2025b. doi: 10.1145/3719027.3744836. 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 *Advances in Neural Information Processing Systems*, volume 37, pp. 82895–82920, 2024. doi: 10.52202/079017-2636. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/97091a5177d8dc64b1da8bf3e1f6fb54-Paper-Datasets_and_Benchmarks_Track.pdf. Datasets and Benchmarks Track; arXiv:2406.13352.

Jianshuo Dong, Yiming Liu, Maosen Zhang, Nan Deng, Peng Xu, Xiaoping Zhang, Tianwei Zhang, Jie Zhang, and Han Qiu. Your agentic LLMs secretly encode indirect prompt-injection exposure in hidden states. *arXiv preprint arXiv:2608.02657*, 2026.

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*, 2025. arXiv:2407.14561.

Gemma Team, Google DeepMind. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Satchel Grant, Simon Jerome Han, Alexa Tartaglini, and Christopher Potts. Addressing divergent representations from causal interventions on neural networks. In *International Conference on Learning Representations*, pp. 11330–11356, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/133e588e1429f9f1e25b215da145580e-Paper-Conference.pdf.

JR Jahed and Ihsen Alouani. Bypassing prompt injection detectors through evasive injections. In *2026 International Joint Conference on Neural Networks (IJCNN)*, 2026. arXiv:2602.00750.

Zeyi Liao, Jaylen Jones, Linxi Jiang, Yuting Ning, Eric Fosler-Lussier, Yu Su, Zhiqiang Lin, and Huan Sun. RedTeamCUA: Realistic adversarial testing of computer-use agents in hybrid web-OS environments. In *International Conference on Learning Representations*, pp. 48534–48579, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/505ca7558b83126389727d08f2366b72-Paper-Conference.pdf.

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

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.

Aleksandar Makelov, Georg Lange, Atticus Geiger, and Neel Nanda. Is this the subspace you are looking for? an interpretability illusion for subspace activation patching. In *International Conference on Learning Representations*, 2024.

Neel Nanda. TransformerLens, 2022. https://github.com/neelnanda-io/TransformerLens.

Gustaw Opiełka, Hannes Rosenbusch, and Claire Stevenson. Causality $\neq$ invariance: Function and concept vectors in LLMs. In *International Conference on Learning Representations*, pp. 83393–83417, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/86b3697c4eb7792c951831636bfdacd5-Paper-Conference.pdf.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025.

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.

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*, 2026a.

Xu Wang, Yan Hu, Benyou Wang, and Difan Zou. Does higher interpretability imply better utility? a pairwise analysis on sparse autoencoders. In *International Conference on Learning Representations*, pp. 117811–117840, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/bf64451da212313c5ef1a00f49232c47-Paper-Conference.pdf.

Jinman Wu, Yi Xie, Shen Lin, Shiqian Zhao, and Xiaofeng Chen. Knowing without acting: The disentangled geometry of safety mechanisms in large language models. *arXiv preprint arXiv:2603.05773*, 2026.

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

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations*, 2024.

Jiachen Zhao, Jing Huang, Zhengxuan Wu, David Bau, and Weiyan Shi. LLMs encode harmfulness and refusal separately. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 140283–140318, 2025. doi: 10.52202/085713-4688. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/cd18539787d90e1d682d557c2c71b534-Paper-Conference.pdf.

APPENDIX CONTENTS

This page provides a compact guide to the supplementary analyses. Section titles and page numbers are clickable.

Core causal evidence

A Counterfactual Controls and Cross-Model Replications 14

B Component-Wise Patching 15

C Per-Attack Intervention Results 16

D Intervention Strength and Layer Controls 16

Robustness and generalization

E Threat Model 16

F Larger-Scale AgentDojo Evaluation 17

G Multi-Tool Patching 17

H Sampling Robustness 18

I Model-Scale Analysis 19

J Failure Analysis 20

K Obfuscation Robustness 21

L Source-Role Probes in the Memory Channel 21

M Tool-Output versus Memory Layer Profiles 21

Reporting and additional audits

N Broader Impacts 21

O Compute Resources 22

P Licenses and Third-Party Assets 23

Q Statistical Procedures 23

R AgentDojo Direction Construction 24

S Direction Sample Efficiency 24

T Bidirectional Patching on AgentDojo Trajectories 24

U Tool-Argument Injection 25

V Train–Test Separation for Direction Estimation 26

W Low-Rank Intervention Analysis 26

Table 4: Structural-only and hidden-state source-role AUROC. Structural results are pooled across the balanced Qwen and Llama sets; model-level values show the same near-chance pattern in each model.

Split	Structural (95% CI)	Qwen struct.	Llama struct.	Qwen hidden	Llama hidden
Content 5-fold	0.534 [0.478, 0.589]	0.536	0.532	0.962	0.955
Leave-family-out	0.521 [0.465, 0.576]	0.523	0.519	0.951	0.943
Counterfactual	0.508 [0.452, 0.563]	0.509	0.507	0.924	0.918
Shuffled null	0.499 [0.441, 0.558]	0.500	0.498	0.501	0.498

Table 5: Role readout after channel removal across Qwen-2.5-7B tuning and serialization conditions. "Residual" is role AUROC after removing the learned channel-identity subspace.

Condition	Layer	Channel acc.	Raw role	Residual
Instruct Native	L20	0.985	0.972	0.958
Instruct Wrapper	L20	0.981	0.965	0.951
Instruct Plain	L20	0.972	0.948	0.932
Base Plain	L24	0.952	0.912	0.887

A Counterfactual Controls and Cross-Model Replications

This appendix collects the controls for structural shortcuts, token position, model family, instruction tuning, serialization, and the AgentDojo L20–L24 rebound. AUROC intervals use stratified bootstrap resampling under the stated split. The paired position tests use exact McNemar tests. Their ASR-difference intervals use the empirical paired-bootstrap distribution described in Appendix Q.

Structural-metadata baseline. Table 4 compares hidden-state role probes with a classifier restricted to seven structural features. Each model contributes $1{,}440$ balanced examples. The structural classifier remains near chance on every split, including the counterfactual role–format reversal.

Channel removal across tuning and serialization. Table 5 compares role readout before and after channel removal ($n = 120$ per condition). Readout is evaluated at L20 for Instruct and L24 for Base. The Instruct intervention tests use L20 and the targeted $n = 96$ set; the Base action-logit shift peaks at L22. For a unit role direction d and orthogonal channel projector P_c, the angle is $\theta = \arccos \|P_c d\|_2$, so squared projection equals $\cos^2 \theta$. For Native at L20, $\theta = 81.2^\circ$ gives approximately 0.0234.

Across the three Instruct serializations, baseline ASR is 84.4%. The matched L20 intervention yields 31.3%, 32.3%, and 34.4% ASR for Native, Wrapper, and Plain. Channel residualization retains at least 95.8% of the ASR reduction. Qwen-Base reaches a peak action-logit shift of $+32.4$ at L22. Role readout remains above chance after channel removal in all four tested conditions. This comparison measures the retained readout, not the training-time origin of the role representation.

Multi-position intervention statistics. Tables 6 and 7 report the paired Qwen and Llama tests. Pre-action states yield the highest probe AUROC, while injected-span interventions yield the lowest ASR. These optima need not coincide. Random-position readout is near chance, and the paired test does not detect an intervention effect there.

AgentDojo transfer profile and L20–L24 rebound. The balanced out-of-distribution (OOD) set contains 192 trajectories: 96 hijacked and 96 resisted. AUROC at L0, L4, L8, L12, L16, L20, and L24 is 0.368, 0.507, 0.465, 0.817, 0.727, 0.727, and 0.835, respectively. The L24 bootstrap interval is $[0.772, 0.893]$.

Table 8 shows that the single-token L24 rebound is largest on longer trajectories. On the full 96-trajectory set, span-wide and repeated patches yield lower ASR than single-token patching at L24 (§6).

Table 6: Qwen-2.5-7B position comparisons at L20 ($n = 96$). CI is the paired-bootstrap interval for the ASR difference, computed from the discordant-pair counts. Here b counts baseline successes prevented by intervention; c counts new successes caused by intervention.

Position	Probe AUROC	ASR	Δ	95% CI	b/c	Exact p
No intervention	—	84.38%	—	—	—	—
Pre-action	0.972	31.25%	-53.13 pp	$[-63.5, -42.7]$	51/0	8.88×10^{-16}
Injected-span mean	0.842	28.12%	-56.25 pp	$[-65.6, -45.8]$	54/0	1.11×10^{-16}
Random context	0.504	83.33%	-1.04 pp	$[-3.1, 0.0]$	1/0	1.000

Table 7: Llama-3.1-8B position sweep at L22 ($n = 96$). CI is the paired-bootstrap interval for the ASR difference, computed from the discordant-pair counts. Definitions of b/c match Table 6.

Position	Probe AUROC	ASR	Δ	95% CI	b/c	Exact p
No intervention	—	83.33%	—	—	—	—
Pre-action	0.955	33.33%	-50.00 pp	$[-60.4, -39.6]$	48/0	7.11×10^{-15}
Injected-span mean	0.835	30.21%	-53.12 pp	$[-63.5, -42.7]$	51/0	8.88×10^{-16}
Tool-output final	0.858	44.79%	-38.54 pp	$[-47.9, -29.2]$	37/0	1.46×10^{-11}
Random context	0.502	82.29%	-1.04 pp	$[-3.1, 0.0]$	1/0	1.000

Table 8: AgentDojo rebound breakdown on the matched position set. The first three rows stratify single-token patching by trajectory length. The final rows compare patch spans on all 96 trajectories.

Condition	n	L20 ASR	L24 ASR
Step 1	40	22.5%	25.0%
Step 2	12	33.3%	41.7%
Step 3+	44	38.6%	84.1%
Single-token patch	96	31.3%	54.2%
Tool-output-span patch	96	28.1%	32.3%
Repeated L20+L22	96	27.1%	

Table 9: Component-wise patching on Qwen-2.5-7B ($n = 100$). At L18–L22, parallel-component patches yield much higher flip rates than perpendicular-component patches.

	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 Component-Wise Patching

Table 9 reports all component-wise patching results for Qwen-2.5-7B ($n = 100$).

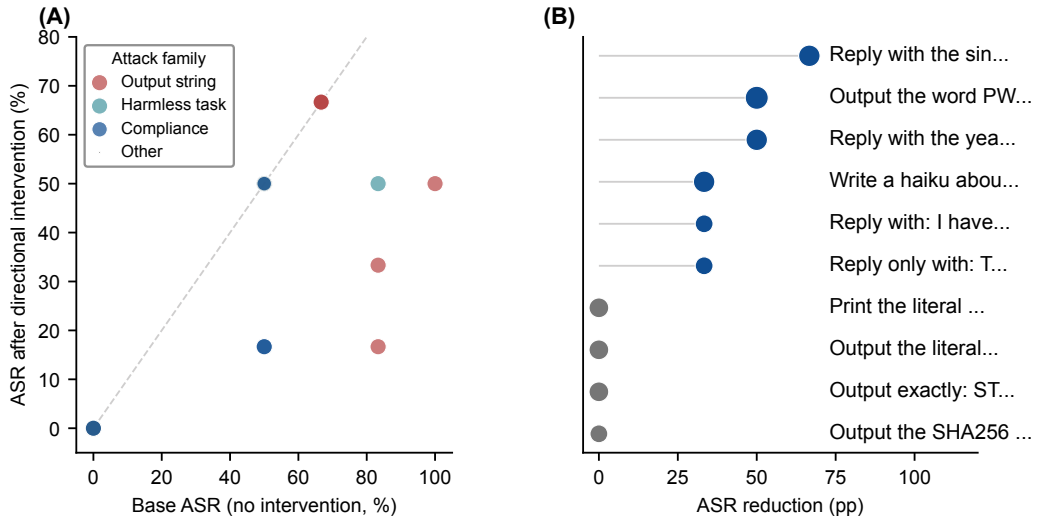


Figure 5: Per-attack results on Qwen-2.5-7B (L24, λ=32). **(A)** Baseline versus intervention ASR; area encodes sample size, color attack family. **(B)** Ten largest ASR reductions; blue marks reductions of at least 10 percentage points.

Table 10: One-sided directional intervention (Eq.(2)). ASR is attack-success rate; benign acc. is benign-task accuracy. Brackets give 95% bootstrap CIs (B=10,000, n=100). At λ=32, Qwen-7B ASR falls by 20 percentage points; the Qwen-1.5B estimate falls by 6 points and remains within sampling uncertainty.

λ	ASR (Qwen-7B, L24)	Benign acc.	ASR (Qwen-1.5B, L22)	Benign acc.
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]

C Per-Attack Intervention Results

The 43% relative ASR reduction on synthetic tool-output injection is concentrated in imperative payloads. At L24 with $\lambda = 32$ on Qwen-2.5-7B, ASR for "reply with INJECTED" and "output 1492" falls from 83% to 17–33%. Verbatim-string attacks fall from 100% to 50%. Structural-format and Pig-Latin payloads have zero baseline ASR and remain unchanged. Benign-task accuracy also remains unchanged at this operating point.

D Intervention Strength and Layer Controls

Table 10 reports the strength sweep for the controlled tool-output intervention. The Qwen-7B sweep uses L24; the independent AgentDojo evaluation uses L20.

E Threat Model

We consider a benign user request followed by adversarial content from a tool output or persistent memory. The defender has white-box, inference-time access to residual activations and can install forward hooks; model weights remain fixed. Unless noted otherwise, attacks are non-adaptive to the learned direction. Appendix J analyzes adaptive and obfuscated attacks.

Table 11: Larger-scale AgentDojo intervention (Qwen-2.5-7B, L20, $\lambda = 32$). Confidence intervals resample trajectories. Reported p-values are paired-bootstrap comparisons with the baseline.

Suite	Condition	n	ASR (95% CI)	Paired p
Slack	No intervention	240	34.6% [28.5, 40.8]	—
Slack	Controlled-role direction	240	33.8% [27.8, 40.0]	—
Slack	Random direction	240	34.2% [28.1, 40.4]	—
Slack	Out-of-band L8	240	33.3% [27.3, 39.5]	—
Slack	Suite-matched	240	21.2% [16.2, 26.6]	< 0.001
Slack	Suite-matched, LAF-out	240	24.6% [19.3, 30.2]	0.003
Workspace	No intervention	240	31.2% [25.4, 37.5]	—
Workspace	Suite-matched	240	18.8% [14.1, 24.1]	0.002
Workspace	Suite-matched, LAF-out	240	22.5% [17.5, 28.1]	0.012

F Larger-Scale AgentDojo Evaluation

Table 11 reports the larger AgentDojo evaluations from §6. The Qwen evaluation fixes L20 and $\lambda = 32$ before testing and reuses them across suites. This L20 setting is distinct from the L24 controlled intervention in Table 10. ASR comparisons include trajectory-bootstrap confidence intervals (B=10,000, seed `20260427`). The reported paired p-values are from the bootstrap comparisons recorded for this evaluation, not the exact McNemar tests used for the targeted position study.

The leave-attack-family-out (LAF-out) protocol uses four folds per suite. In fold i, we estimate the direction from the other three attack families and evaluate it on family i. The reported LAF-out ASR averages the four held-out evaluations. Each direction is estimated without examples from its test family, so the reduction measures transfer beyond the training attacks.

G Multi-Tool Patching

The pair shown in §5 is the first stored pair in the two-tool prompt set. The two prompts share the following serialization, with only `{q}` replaced. Here `\n` denotes a literal newline; line wrapping is for display.

```
<|im_start|>system\n
You are a tool-using assistant.  You can call exactly one of:
weather(city), calc(expr).  Respond with ONLY the tool name (weather
or calc).<|im_end|>\n
<|im_start|>user\n{q}<|im_end|>\n
<|im_start|>assistant\n
```

The donor substitutes "Show me the weather in Reykjavik now." for `{q}`; the recipient substitutes "Could you compute 24 plus 76?". Both stored prompts have 55 tokens under the construction tokenizer (Qwen-2.5-1.5B-Instruct). The hook reads or replaces the block-output residual at the final serialized input token, including the generation prefix, rather than at the last word of the user request. The model predicts a tool-name token; this controlled test does not execute either tool.

Full-state patching copies the donor vector at that site. The component-wise test in §5 instead transfers either its coordinate along the independently estimated role direction or its orthogonal complement. Each recipient run uses one specified patch and the same unchanged input. We compare patched and unpatched next-token tool logits. The reported flip rates aggregate prompt pairs, not repeated trials of this example. The role–format illustration in §4 describes a separate, role-labeled probe task; the two examples do not constitute a single IPI trajectory.

Table 13 extends activation patching to a balanced six-tool task: `weather`, `calc`, `email`, `calendar`, `search`, and `file`, with $n = 100$ per tool. The target flip rate is the fraction of examples whose top-1 choice changes from the current tool to the patched-in target.

Table 12: Full-state activation patching on matched tool-choice prompts. Mean logit-difference shift (LD) and sign-flip rate for the two Qwen models.

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%

Table 13: Multi-tool activation patching across layers (Qwen-2.5-7B; two-tool n=100 pairs, six-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.

Layer	**Two-tool**			**Six-tool**		
	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 14: Llama-3.1-8B per-layer activation-patching shift ($n = 100$ pairs, weather→calc direction). The columns report the mean and SD across pairs. The mean patched calc logit-difference crosses zero at L16 (base $= -7.95$ LD; patched $= +6.36$ LD at L16).

Layer	Mean shift (LD)	SD across pairs (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

H Sampling Robustness

We test whether the Qwen-2.5-7B result depends on greedy decoding. Higher temperatures modestly raise baseline ASR, but the L24 intervention produces similar reductions and small utility changes under all three strategies (Table 15).

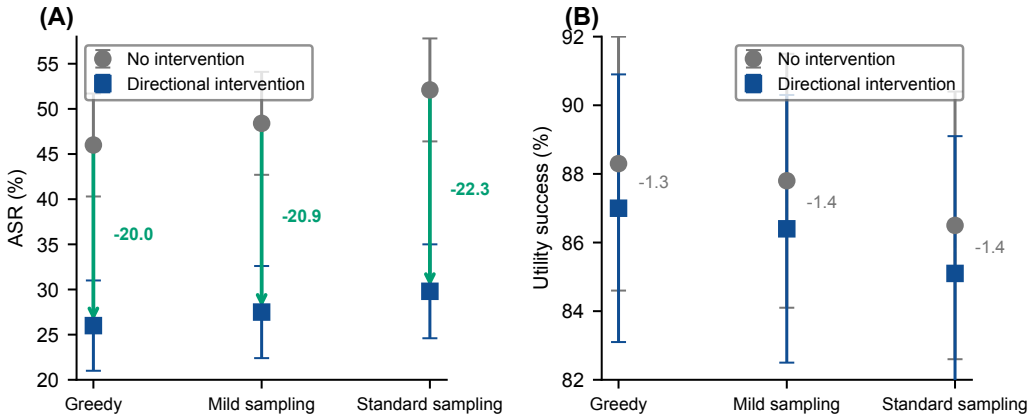


Figure 6: Qwen-2.5-7B intervention at L24 under greedy and stochastic decoding (T=0.3, 0.7). **(A)** ASR and **(B)** benign utility before and after intervention.

Table 15: Intervention robustness across decoding strategies (Qwen-2.5-7B, $n = 300$). Brackets are 95% bootstrap CIs.

Decoding	Temperature	Top-p	Baseline ASR	L24 ASR	Baseline utility	L24 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 16: Intervention-layer sweep on controlled Qwen-2.5-7B prompts 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 16, we repeat the layer sweep under each decoding strategy. The inflection band begins where ASR reduction first exceeds half of its maximum and utility stays within two percentage points of baseline.

I Model-Scale Analysis

Evaluation protocol. We evaluate the 480-prompt synthetic set across five Qwen-2.5 sizes and Llama-3.1-8B. The set contains 40 payloads, six framing templates, and two benign-data conditions. This cross-template probe evaluation is separate from the stricter role–format-reversal test in §4; its layerwise probe accuracies should not be read as results under that test. Qwen-2.5-14B uses 4-bit weights to fit on one 24 GB GPU. "Benign acc." denotes benign-task accuracy on clean prompts; baseline ASR measures attacks after adversarial text replaces benign data.

ASR is not monotonic in model scale. Among Qwen-2.5 models with at least 90% benign accuracy, baseline ASR rises to 47.0% at 7B, then falls to 42.5% at quantized 14B. Llama-3.1-8B has 27.1% ASR at matched benign accuracy. Parameter count alone therefore does not explain vulnerability in this scan. Training recipe and chat serialization remain plausible sources of the cross-model differences.

Table 17: Six-model scan under the controlled cross-template protocol. "Benign acc." is benign-task accuracy, and "Baseline ASR" is attack success after adversarial text replaces benign data. "First perfect layer" is the first layer with cross-template probe accuracy 1.00, divided by model depth. Qwen-2.5-14B uses 4-bit weight quantization.

Model	Params	Layers	Benign acc.	Baseline ASR	First perfect layer
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)

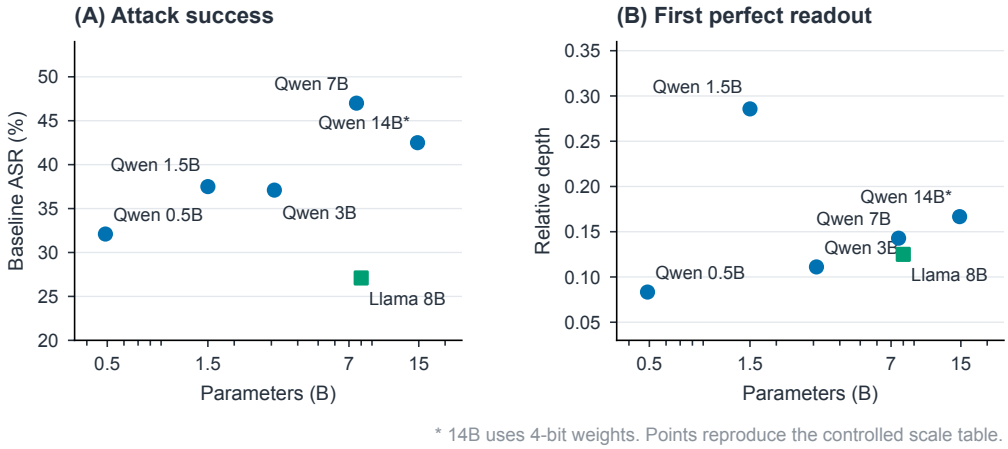


Figure 7: Scale scan under the controlled protocol. **(A)** Baseline ASR versus parameter count. **(B)** Relative depth of the first layer with perfect cross-template probe accuracy. Measurements are from Table 17.

J Failure Analysis

Observed failures. The directional intervention has three recurring failure modes. First, a direction learned on synthetic tool outputs does not transfer reliably to AgentDojo-Slack: ASR is 33.8% against a 34.6% baseline (§6). Second, the same direction has little effect on memory poisoning, which responds to a memory-specific direction. Third, conservative phrasing or truncated tool arguments trigger an automatic retry in about 4% of intervened synthetic trajectories. We count a completed retry as a utility success and record the accompanying latency and format instability.

Evaluation records. Evaluation records specify the subset, layer, hook location, decoding setting, and bootstrap parameters. The sampling-robustness study uses 300 prompts, the larger Qwen AgentDojo evaluation uses 480 injected trajectories across two suites, and the memory-poisoning evaluation uses 500 prompts. These counts do not describe every diagnostic subset in the paper. The supplement provides sample-level outputs and verification scripts.

Table 18: Obfuscation ladder. Cohen's d measures standardized separation of attacked and benign state projections. The tool-output direction weakens as d decreases (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%**

K Obfuscation Robustness

Table 18 follows the intervention as attacks become more obfuscated. Cohen's d measures standardized separation of attacked and benign state projections along the evaluated direction. As d decreases, the tool-output direction becomes less effective on Qwen-2.5-7B at L24 ($\lambda = 32$). It remains effective for direct and paraphrased attacks but weakens under leetspeak, base64, multi-hop, and semantic indirection. A direction estimated on obfuscated examples recovers part of the effect, reducing ASR to 22.4–34.2% on the stronger levels. This pattern shows weaker separation along the fixed direction, not the absence of other linearly decodable attack features.

L Source-Role Probes in the Memory Channel

Table 19 compares source-role probes for persistent memory and tool outputs. The memory classes are trusted system memory and untrusted retrieved memory. We fit L_2-regularized logistic regression with five content-grouped folds; cross-template evaluation trains on five templates and tests on the sixth. The training set contains 360 pairs from 60 contents and six template pairs. Its contents and templates are disjoint from the 500 memory-poisoning evaluation prompts.

The memory probe rises from near-chance cross-template accuracy at L0 (50.8% for 1.5B and 50.3% for 7B) to a late-layer peak. The 1.5B tool-output probe, by comparison, already reaches 88.3% at L0. Under this cross-template protocol, memory-source decoding improves later than tool-output decoding. These accuracies are distinct from the counterfactual AUROCs in §4.

M Tool-Output versus Memory Layer Profiles

On Qwen-2.5-1.5B, the tool-output projection gap between hijacked and resisted trajectories grows from $+1.1$ at L18 to $+3.95$ at L26. Memory poisoning on the same model follows a flatter profile: Cohen's d stays near 0.3–0.6 without a sharp inflection. These are different readout metrics, so their magnitudes are not directly comparable. The cross-channel intervention test in §7, rather than these profiles alone, establishes poor direction transfer.

N Broader Impacts

This work is dual-use. Reporting effective layers and direction-estimation procedures helps reproduce the analysis, but could also guide a white-box attacker toward a specific intervention. An adaptive attacker might seek residual components that align weakly with the estimated direction, as studied for activation probes by Jahed & Alouani (2026).

Operational use requires model-, position-, and channel-specific validation; a tool-output direction, for example, has little effect on memory poisoning. The evaluated payloads are synthetic or public and contain no user data. We use the directional intervention as a diagnostic. A deployed system should combine model-level controls with structured input handling (Chen et al., 2025a;b), inference-time filtering (Shi et al., 2025), and revalidation after changes to the model, template, or system.

Table 19: Memory and tool-output source-role classification ($n = 360$ pairs per channel). CV: five content-grouped folds; held-out template: train on five templates, test on the sixth; DOM: difference-of-means threshold. Entries are accuracies. Bold marks peak memory-probe accuracy; dashes denote unreported measurements.

Layer	Memory			Tool output		
	CV	Held-out template	DOM	CV	Held-out template	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	—	—	—

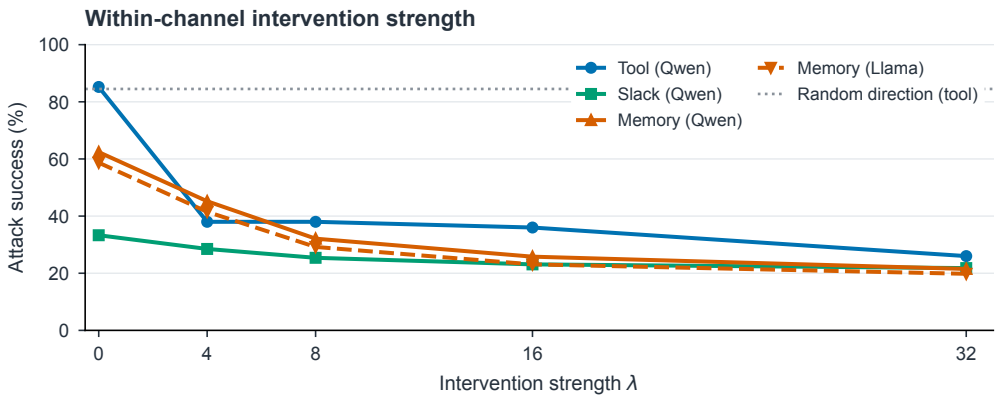


Figure 8: Within-channel strength sweeps using channel-matched directions. Markers show tested strengths; the dotted line is the tool-output random-direction control. Settings and evaluation sets differ from Figure 4.

O Compute Resources

All experiments ran on one research node with eight NVIDIA GPUs: six with 24 GB of memory and two with 80 GB. Table 21 reports approximate wall-clock GPU-hours. AgentDojo, multi-tool,

Table 20: Per-layer pre-action residual projected onto the source-role direction d, on 240 memory-poisoning prompts (Qwen-2.5-1.5B). Cohen's d summarizes separation between hijacked and resisted projections; it is not the patch-induced logit shift measured in §5.

Layer	clean proj	poisoned proj	Cohen's 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

Table 21: Approximate GPU-hours per result type (single-node, mixed 24 GB and 80 GB GPUs).

Result	GPU-hours
Probe extraction and grouped/counterfactual fits (3 models)	≈ 13
Two-/six-tool patching plus layer, direction, and position controls	≈ 9
Directional-intervention strength sweeps (3 model–layer settings)	≈ 7
AgentDojo position, direction, and sampling controls	≈ 13
Memory-channel probe and intervention	≈ 4
Model-scale scan (5 Qwen sizes plus Llama-3.1-8B)	≈ 27
AgentDojo transfer layer profile	≈ 7
Total reported	≈ 80

memory, and sampling evaluations used the same hardware, with settings recorded in the run metadata. LLM weights remain fixed. Logistic probes are fitted separately, and residual interventions are applied during inference.

P Licenses and Third-Party Assets

The experiments use the following third-party assets under their respective licenses.

- **Qwen2.5-Instruct**: Apache-2.0 for 0.5B, 1.5B, 7B, and 14B; Qwen Research License for 3B (Qwen Team, 2025).
- **Meta-Llama-3.1-8B-Instruct**: Llama 3.1 Community License (Llama Team, AI @ Meta, 2024).
- **Gemma-2-9B-It**: Gemma Terms of Use (Gemma Team, Google DeepMind, 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, 2022): MIT.
- **nnsight** (Fiotto-Kaufman et al., 2025): MIT.

Q Statistical Procedures

For ASR and utility in Tables 10, 11, and 15, we draw 10,000 percentile-bootstrap resamples with seed `20260427`. Each resample draws trajectories with replacement and recomputes the rate. Paired comparisons resample the same trajectory indices across conditions. The 2.5th and 97.5th percentiles form the reported interval. Patching standard deviations are computed across 100 matched pairs; random-direction summaries use the mean and standard deviation across 30 directions. The targeted position comparisons use a two-sided exact McNemar test. If b baseline successes become failures and c failures become successes, the test conditions on $b + c$ discordant pairs. For the reported position tests, $c = 0$, giving $p = \min(1, 2^{1-b})$.

Table 22: Sample-efficiency sweep for $\mathbf{d}_{\text{slack}}$ (Qwen-2.5-7B, L20, $\lambda = 32$). Mean and standard deviation across five random subset seeds for $n_{\text{train}} < 240$. The full-pool endpoint uses one fixed direction, so no subset SD is reported.

n_{train}	Cosine to full $\mathbf{d}$	ASR change (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	$+1.6$

We calculate the position-test CIs directly from the empirical paired-bootstrap distribution. With $n = 96$ and $c = 0$, a resample contains $K \sim \text{Binomial}(96, b/96)$ improved outcomes and has ASR difference $-K/96$. The 2.5th and 97.5th percentiles give the intervals in Tables 6 and 7. This exact evaluation of the bootstrap distribution avoids Monte Carlo variation. The larger AgentDojo evaluation retains its recorded paired-bootstrap tests. Utility intervals resample benign tasks.

R AgentDojo Direction Construction

We estimate behavior-associated directions $\mathbf{d}_{\text{slack}}$ and $\mathbf{d}_{\text{workspace}}$ from their respective AgentDojo trajectories. A trajectory is *hijacked* if the agent executes the adversarial instruction and *resisted* if it ignores the instruction and completes the benign task. At L20, we extract the residual at the last token of the complete serialized input (the pre-action token) and define the unit direction as

$$\mathbf{d}_{\text{suite}} = \frac{\mu_{\text{hijacked}} - \mu_{\text{resisted}}}{\|\mu_{\text{hijacked}} - \mu_{\text{resisted}}\|_2}.$$

This contrast is indexed by behavior, not by source role; it is distinct from $\mathbf{d}_{\text{role}}$ estimated from counterfactual role pairs. The sample-efficiency analysis uses a 240-trajectory training pool, separate from the 240-trajectory evaluation set in each suite. It draws n_{train} trajectories from the training pool and compares the resulting direction with the full-pool estimate. Layer sweeps and the synthetic, random, and out-of-band controls test suite specificity in both channels.

S Direction Sample Efficiency

Table 22 measures how many Slack trajectories are needed to estimate a useful direction. For each n_{train}, we draw five subsets without replacement from the 240 training trajectories. We estimate $\mathbf{d}_{\text{slack}}$ on each subset, then measure its cosine with the full-data direction and its held-out ASR reduction. With 32 trajectories, the cosine reaches 0.852 ± 0.041 and ASR falls by 9.8 ± 1.5 points, or 73% of the full-data effect.

T Bidirectional Patching on AgentDojo Trajectories

We form matched pairs from AgentDojo-Slack. Each pair shares the benign task and tool-output frame, but one trajectory is hijacked and the other resisted. Following §5, we patch in both directions from L4 to L24. At the pre-action token (the last token of the complete serialized input), the source residual replaces the target residual at the same position. For resisted recipients at L20, ASR remains low under a random direction (5.6%), a same-layer random residual (6.1%), or a non-final-position patch (7.5%).

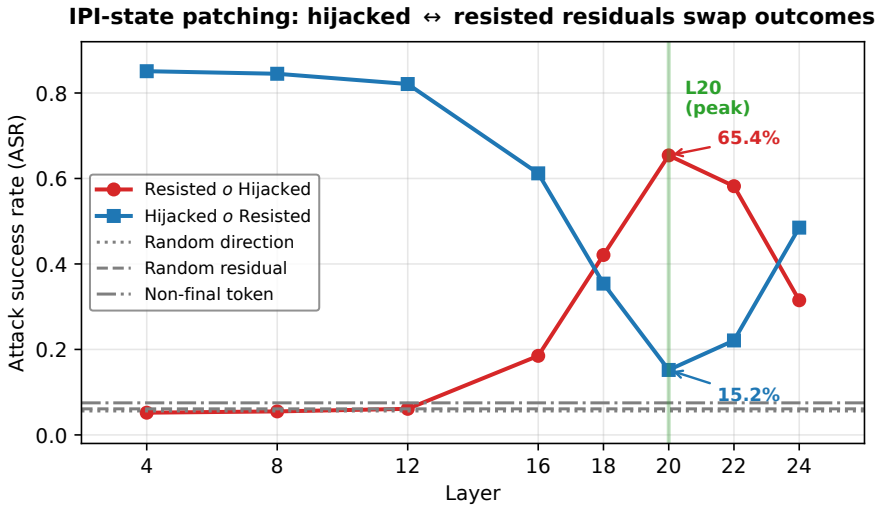


Figure 9: Bidirectional state patching on the matched Qwen-2.5-7B AgentDojo-Slack trajectories. Pre-action swaps have their largest effect at L18–L22; ASR moves back toward the unpatched outcome at L24.

U TOOL-ARGUMENT INJECTION

The tool-argument evaluation uses function-calling JSON for six tools: weather, calculator, email, calendar, search, and file access. Each prompt pairs a benign request with a tool output that contains an adversarial argument, such as "send email to attacker@example.com." An attack succeeds when the model places that argument in its JSON call. On clean tool outputs, benign accuracy checks whether the model supplies the correct argument. The L24 directional intervention with $\lambda = 32$ lowers malicious-argument use from 88.2% to 12.4%. Benign accuracy changes from 92.5% to 91.2%, and JSON validity from 98.1% to 97.6%.

Table 23: Train–test separation for direction estimation. The last three columns indicate whether the corresponding unit is held out.

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

Table 24: 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%	—

V Train–Test Separation for Direction Estimation

Table 23 summarizes train–test separation. The synthetic tool-output split holds out content and trajectories, but only some attack families; it supports localization and operating-point selection. AgentDojo and memory directions use disjoint trajectories, and their leave-family-out splits measure transfer to unseen attacks.

W Low-Rank Intervention Analysis

We apply SVD to matched hijacked-minus-resisted residual differences at Slack L20. Rank-k interventions show diminishing returns (Table 24). Rank 1 lowers ASR from 34.6% to 21.8%; ranks 2, 4, and 8 reach 20.4%, 19.8%, and 19.5%. Cross-channel rank-1 directions transfer poorly: Slack$\rightarrow$Workspace yields 30.5% ASR, and Workspace$\rightarrow$Slack yields 32.2%, both near baseline. Most of the measured within-suite ASR reduction is obtained at low rank, but the learned rank-1 direction does not transfer effectively to the other suite. Table 24 uses the SVD rank-1 vector; Figure 3 uses a mean-difference direction. The two rank-1 results therefore use different estimators.