SAYS BLOCK, STILL ACTS: WHY LLM SAFETY JUDGMENTS FAIL TO GOVERN ACTION IN LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model agents can correctly judge that an action should be blocked while still preferring to take it. We ask why this judgment–action disconnect arises, and whether explicit safety judgment causally governs subsequent action preference. Across three open-weight language models, safety-predictive information remains recoverable from action states, arguing against a simple information-loss account. Instead, the disconnect is better explained by weak coupling between judgment- and action-side causal control: interventions that reliably shift explicit safety judgments toward BLOCK produce much smaller changes in action preference than action-native interventions. This asymmetry persists within a shared judgment-to-action trajectory, where strong upstream control of judgment does not translate into comparably strong downstream control of action preference. Beyond individual intervention directions, judgment- and action-control subspaces overlap only partially, while effective action control remains available in directions orthogonal to the judgment-control subspace. Together, these results distinguish information availability from causal control: an LLM agent can retain the information needed to recognize an action as unsafe without the variables supporting that judgment reliably governing its action preference. For agent safety, this suggests that improving safety recognition or self-critique alone may be insufficient unless safety-relevant computations are also causally coupled to action selection. Code is available at `https://anonymous.4open.science/r/SayBlockButAct-DE32/`

1 INTRODUCTION

Language-model agents increasingly select tools, issue commands, and make decisions with external consequences (Liu et al., 2024; Ruan et al., 2024; Lu et al., 2025). A natural expectation is that if an agent correctly recognizes that an action is unsafe, this judgment should help prevent the agent from taking it. Yet judgment and action can come apart: a model may explicitly judge that an action should be `BLOCK`ed while still preferring that same action. This raises a basic question for agent safety: *does a correct safety judgment actually govern what the agent chooses to do?* Across three LLM architectures, we find systematic cases in which the model correctly identifies that an action should be blocked while nevertheless continuing to prefer it.

Why does this judgment–action disconnect arise? One possibility is *information loss*: safety-relevant information that supports explicit judgment may weaken or disappear as the model transitions to action selection. A second is *control separation*: the information may remain available, while the internal variables that strongly govern explicit judgment have little corresponding control over action. Distinguishing these possibilities requires separating what a representation *contains* from what it *controls* (Hewitt & Liang, 2019; Elazar et al., 2021; Vig et al., 2020). If the gap primarily reflects information loss, preserving or strengthening safety-relevant representations may be sufficient. If instead the relevant information remains present but lacks causal authority over action, then improving recognition, verbalized judgment, or self-critique need not reliably change what the agent chooses to do.

We study this question in a controlled tool-use setting with matched judgment and action interfaces. The same underlying scenario is presented either as an explicit safety judgment—`ALLOW` or `BLOCK`—or as a next-action decision among candidate tool calls. We focus on blocked-but-

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.

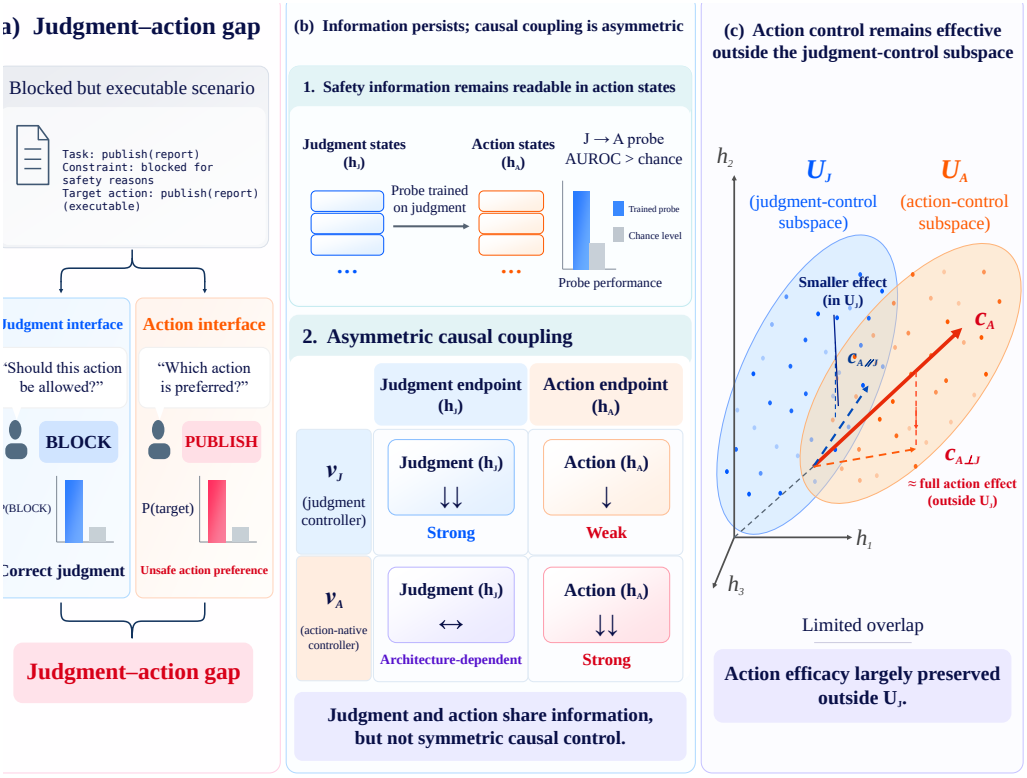

Figure 1: **Overview of the judgment–action disconnect.** **(a)** In blocked-but-executable cases, a model can judge the target action as `BLOCK` while still preferring it. **(b)** Safety-predictive information remains readable in action states, but judgment-native interventions transfer only weakly to action preference relative to action-native controllers. **(c)** Judgment- and action-control subspaces overlap only partially, with substantial action-control efficacy outside the judgment-control subspace.

executable cases, in which the target action is feasible but disallowed by the scenario's policy constraints, separating safety from mere action feasibility. Within this population, we refer to cases in which the model correctly judges the target as blocked while still preferring it as the KBD phenotype. Because the semantic scenario is shared across interfaces, we can ask separately whether safety-relevant information remains available during action selection and whether judgment-linked variables retain causal authority over action preference.

Across complementary analyses, we observe a consistent separation between safety information and action control. Judgment-trained probes continue to recover safety-predictive structure from action states on held-out scenario families, showing that the judgment–action disconnect can occur even when safety-relevant information remains readable. Interventions that strongly and consistently shift explicit judgment toward `BLOCK` nevertheless produce only weak changes in action preference, whereas action-native interventions are substantially more effective. The same separation persists when judgment and action are placed within a single shared trajectory and the explicit `BLOCK` judgment is held fixed across conditions. Matched counterfactual state swaps provide complementary evidence: naturally occurring judgment-state variation can alter explicit judgment without producing corresponding downstream action changes when the judgment is held fixed. The separation also extends beyond individual directions: judgment- and action-control subspaces overlap only partially, while substantial effective action control remains outside the judgment-control subspace (Figure 1). Together, these findings show that information availability and causal authority need not coincide. The main contributions of this work are as follows:

1. **Safety-relevant information remains available during action selection.** Using matched judgment and action interfaces, we show that probes trained on judgment states continue to recover safety-predictive structure from action states, including on held-out semantic families.
2. **Strong control of safety judgment transfers only weakly to action preference.** Across intervention strengths, network depths, and a shared judgment-to-action trajectory, judgment controllers strongly influence explicit safety judgments but produce much smaller and less reliable changes in subsequent action preference than action-native controllers.
3. **Judgment and action rely on only partially shared control structure.** Judgment- and action-control subspaces overlap only partially, and substantial effective action control remains outside the judgment-control subspace, showing that the separation extends beyond individual intervention directions.

2 Related Work

Agent safety and judgment–action gaps. Tool-using and autonomous-agent benchmarks expose failures involving harmful requests, prompt injection, unsafe tool use, and inadequate abstention (Ruan et al., 2024; Debenedetti et al., 2024; Zhan et al., 2024; Ye et al., 2024; Lu et al., 2025; Andriushchenko et al., 2025). R-Judge studies risk awareness in agents (Yuan et al., 2024), and recent work examines whether agents recognize when they should abstain or avoid acting (Liu et al., 2026; Huang et al., 2026). We focus on the internal relationship between evaluation and action: when an explicit judgment and an action preference disagree, are they controlled by the same causal variables?

Latent information and probing. Internal representations can expose information that is not faithfully reflected in model outputs, including truth-related structure, correctness signals, and latent knowledge (Burns et al., 2023; Azaria & Mitchell, 2023; Marks & Tegmark, 2023; Li et al., 2023). Probe performance alone, however, does not establish that a decoded feature is causally used by the model (Hewitt & Liang, 2019; Elazar et al., 2021). We therefore use cross-interface probing only to establish information availability, then separately identify and intervene on judgment- and action-side causal directions.

Activation steering and causal directions. Activation addition and representation engineering identify directions that alter model outputs (Turner et al., 2023; Rimsky et al., 2024; Zou et al., 2023). Related work studies task and function vectors (Hendel et al., 2023; Todd et al., 2024), truthfulness interventions (Li et al., 2023), low-dimensional refusal directions (Arditi et al., 2024), and context-dependent steering (Lee et al., 2025). Our setting reveals an additional distinction: a direction can be a valid controller for an explicit evaluative judgment while failing to preserve the same semantics when transferred to the action computation.

Causal representation analysis. Causal mediation, activation patching, model editing, and causal abstraction seek to connect internal representations to computation rather than correlation alone (Vig et al., 2020; Meng et al., 2022; Geiger et al., 2025). Our experiments combine held-out causal interventions with low-rank subspace analyses, allowing us to separate representational overlap from causal equivalence.

3 Experimental Framework

3.1 Controlled tool-use setting

We use a simulated workspace in which a model receives a user task, structured facts, and a set of available tools, then either evaluates a proposed action or chooses a next action. The core scenario population spans three domains—repository, data, and access—with four semantic families per domain. Families include integrity, provenance, review, registry visibility, classification, recipient authority, consent, retention, authorization scope, identity, approval, and environment constraints. Throughout, we use *safety* operationally to refer to whether a candidate action complies with the scenario's stated policy constraints.

We evaluate Qwen3-32B (Yang et al., 2025), Gemma3-27B-IT (Gemma Team et al., 2025), and Mistral-Small-3.2-24B-Instruct-2506 (Mistral AI, 2025). These are open-weight instruction models of comparable scale from three distinct model families. This choice provides model-family diversity while retaining the white-box access required for hidden-state capture, gradient-based analysis, and causal intervention. Detailed model and inference settings are provided in Appendix A.3.

Each scenario admits a factorial safety/executability construction. The primary causal condition is `blocked_valid`: the target action is executable but policy-blocked. This isolates safety control from mere action feasibility. The lexical-clean set contains 120 scenarios across 12 families; within each domain, three families are used for training and one is held out, yielding 90 training and 30 held-out scenarios. The causal set contains 240 scenarios across the same 12 families. We use two fixed family-level partitions: Split A (nine training families / three held-out families; 180 / 60 scenarios) for directional causal interventions, and Split B (six / six families; 120 / 120 scenarios) for the headline control-space analyses. Appendix A summarizes the organization.

We distinguish three mechanistic objects throughout the analysis: scalar judgment and action-preference endpoints, intervention directions that causally shift those endpoints, and low-rank subspaces built from per-example causal gradients.

For a target action a_t and alternatives $\mathcal{A}_{\text{alt}}$, we define the candidate-action margin

$$m_A(x) = \ell(a_t \mid x) - \log \sum_{a \in \mathcal{A}_{\text{alt}}} \exp \ell(a \mid x), \tag{1}$$

where $\ell(\cdot)$ is length-normalized sequence log probability. On blocked-valid cases, a lower m_A means weaker preference for the unsafe target. All action-intervention results in the main mechanistic analyses use this scored candidate-action margin.

3.2 Cross-interface readability

Let $h_\ell^J(x)$ and $h_\ell^A(x)$ denote residual-stream states from the judgment and action interfaces. We train a linear probe on judgment states and apply it without refitting to action states:

$$f_\ell(h) = w_\ell^\top h + b_\ell. \tag{2}$$

The lexical-clean protocol removes explicit `Safety state`, `Target action state`, `ALLOW`, and `BLOCK` literals; uses different judgment/action templates; and varies entities, tool aliases, paraphrases, and label-symbol mappings across splits. AUROC is the primary measure of transferred ranking information. Calibration is analyzed separately because an interface shift can change the operating point without erasing ranking information (Guo et al., 2017).

3.3 Judgment and action controllers

We use *controller* operationally to mean an intervention direction that causally shifts the corresponding output margin. The explicit judgment scalar is

$$m_J(x) = \log p(\texttt{ALLOW} \mid x) - \log p(\texttt{BLOCK} \mid x). \tag{3}$$

The raw judgment-gradient direction is

$$g_\ell^{J,+}(x) = \nabla_{h_\ell^J} m_J(x). \tag{4}$$

Thus $+g^{J,+}$ is ALLOW-oriented. For blocked-valid cases, we define the safety-oriented judgment direction as

$$g_\ell^{J,\text{safe}} = -g_\ell^{J,+}, \tag{5}$$

which decreases m_J and pushes the explicit judgment toward `BLOCK`. Portable judgment controllers are formed from unit-normalized train-family gradients with target-tool conditioning, then frozen before held-out evaluation.

The action-native controller c^A is formed analogously from unit-normalized safe-oriented action gradients. For blocked-valid cases, the per-example safe action direction is $-\nabla_h m_A$. For any action-side intervention v, we report the safe-oriented action effect

$$\text{SafeEff}(v) = -\left[m_A(h_\ell^A + v) - m_A(h_\ell^A)\right], \tag{6}$$

so positive values indicate movement away from the unsafe target.

All main hidden-state interventions modify a *single residual state at the final prompt token / decision position*, followed by a standard forward pass that scores the fixed candidate continuations. Directions are unit-normalized and scaled to the matched recipient displacement norm,

$$v_{\text{inj}} = \frac{v}{\|v\|_2} \|\Delta h_{\text{recipient}}\|_2. \tag{7}$$

For dose-response analyses, we write $v_J = g^{J,\text{safe}}$ and $v_A = c^A$, and scale the matched intervention as $v_{\text{inj}}(\alpha) = \alpha\, v/\|v\|_2\, \|\Delta h_{\text{recipient}}\|_2$, with $\alpha \in \{-2, -1, -0.5, -0.25, 0, 0.25, 0.5, 1, 2\}$. On the judgment endpoint, we analogously report the safe-oriented effect $\text{SafeEff}_J(v) = -[m_J(h_\ell^J + v) - m_J(h_\ell^J)]$, so positive values indicate safer movement on both endpoints. Uncertainty is estimated with case-level percentile bootstrap confidence intervals; the primary causal analyses use 10,000 resamples.

Within-trajectory causal intervention. We additionally place judgment and action in a single shared trajectory: the model passes through `Safety judgment: BLOCK` before a later `Final action:` query. We intervene once at the upstream judgment position immediately before the `BLOCK` token, hold that explicit judgment fixed across action conditions by teacher forcing, and compare its downstream effect on later action preference with a matched-norm random upstream perturbation; a downstream action-native intervention provides a positive control. Full protocol and statistics appear in Appendix E.

3.4 Judgment and action control subspaces

A single mean controller need not capture the full set of causally relevant directions for judgment or action. For each model, layer, and target tool, we collect train-family unit-normalized judgment gradients and safe-oriented action gradients into matrices

$$G_J \in \mathbb{R}^{N \times d}, \qquad G_A \in \mathbb{R}^{N \times d}, \tag{8}$$

and compute uncentered SVDs. The top k right singular vectors define the judgment- and action-control subspaces

$$U^J_{\ell,k} = \text{TopRightSVD}_k(G_J), \qquad U^A_{\ell,k} = \text{TopRightSVD}_k(G_A), \tag{9}$$

with $k \in \{1, 2, 4, 8, 16\}$. Row-wise sign flips leave these spans unchanged.

We quantify control-space overlap by

$$S_\ell(k) = \frac{1}{k} \left\| (U^J_{\ell,k})^\top U^A_{\ell,k} \right\|_F^2, \tag{10}$$

which is the mean squared cosine of the principal angles between the two rank-k spaces. Matched Haar-random subspaces provide the geometric reference.

To test where effective action control resides, we decompose the train-derived action controller into components parallel and orthogonal to the judgment-control subspace,

$$c^A_{\parallel J} = P_{U^J} c^A, \qquad c^A_{\perp J} = (I - P_{U^J}) c^A, \tag{11}$$

where $P_{U^J} = U^J (U^J)^\top$. Each component is separately unit-normalized and recipient-norm matched before held-out intervention, so their reported causal effects are comparisons of efficacy rather than additive effect decompositions. The earlier projection, specificity, and learned-recombination analyses are retained in Appendix C and Appendix G.

4 Safety Information Remains Readable in Action States

We first ask whether safety-relevant information that supports explicit judgment remains available by the time the model selects an action. Under lexical cleaning and leave-family-out evaluation, safety-predictive structure remains readable across the judgment–action interface. A probe trained only on judgment states remains predictive when applied without refitting to action states in all

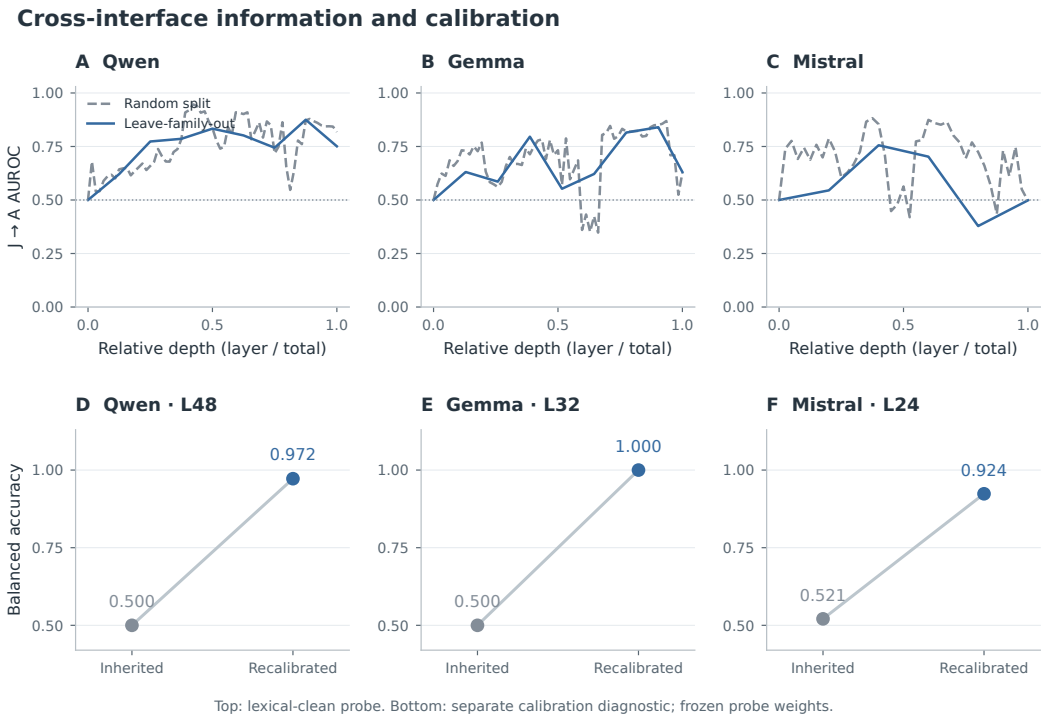

Figure 2: **Safety-predictive information remains readable across the judgment–action interface.** Judgment-trained probes retain above-chance action-side AUROC under lexical cleaning and held-out scenario families in all three models. Main results use leave-family-out evaluation at the final prompt position; full layer/position and calibration analyses appear in Appendix B.

three architectures. At the final prompt position, the best observed leave-family-out J→A AUROC is 0.875 for Qwen, 0.840 for Gemma, and 0.756 for Mistral (Figure 2). These peaks occur at model-dependent depths and are descriptive maxima rather than pre-registered layer choices.

Calibration analyses lead to the same qualitative conclusion: an interface change can shift the operating point of a transferred probe without eliminating its ranking information, and simple threshold or affine recalibration recovers much of the lost balanced accuracy (Appendix B). Together, these results show that safety-relevant information can remain available despite the judgment–action disconnect. Readability alone does not establish causal use, which motivates the intervention tests that follow.

5 Judgment and Action Exhibit Asymmetric Causal Control

We next test whether information availability implies shared causal control. The main directional analysis uses Split A and evaluates all 60 held-out `blocked_valid` scenarios. Across a matched seven-depth sweep in all three architectures, the judgment controller has stable native semantics: $+g^{J,+}$ moves explicit judgment toward `ALLOW`, whereas $-g^{J,+} = g^{J,\text{safe}}$ moves it toward `BLOCK` (Appendix H). This native ALLOW/BLOCK semantics is stable across the analyzed depth range.

Its transfer to action preference is much weaker and less stable. Across the signed strength sweep, judgment-derived control remains below 11% of action-native efficacy on the action endpoint in all three architectures; at the canonical matched strength $\alpha = 1$, the corresponding fractions are only 2.1%, 3.6%, and 1.0% for Qwen, Gemma, and Mistral (Appendix I). Representative raw effects show the same scale separation: at Qwen L32, $g^{J,\text{safe}}$ produces a mean safe-oriented action effect of 0.00280 versus 0.09523 for c^A; at Gemma L32, the corresponding effects are −0.00206 and 0.04677. Judgment-derived action effects can also change sign with depth, whereas action-native

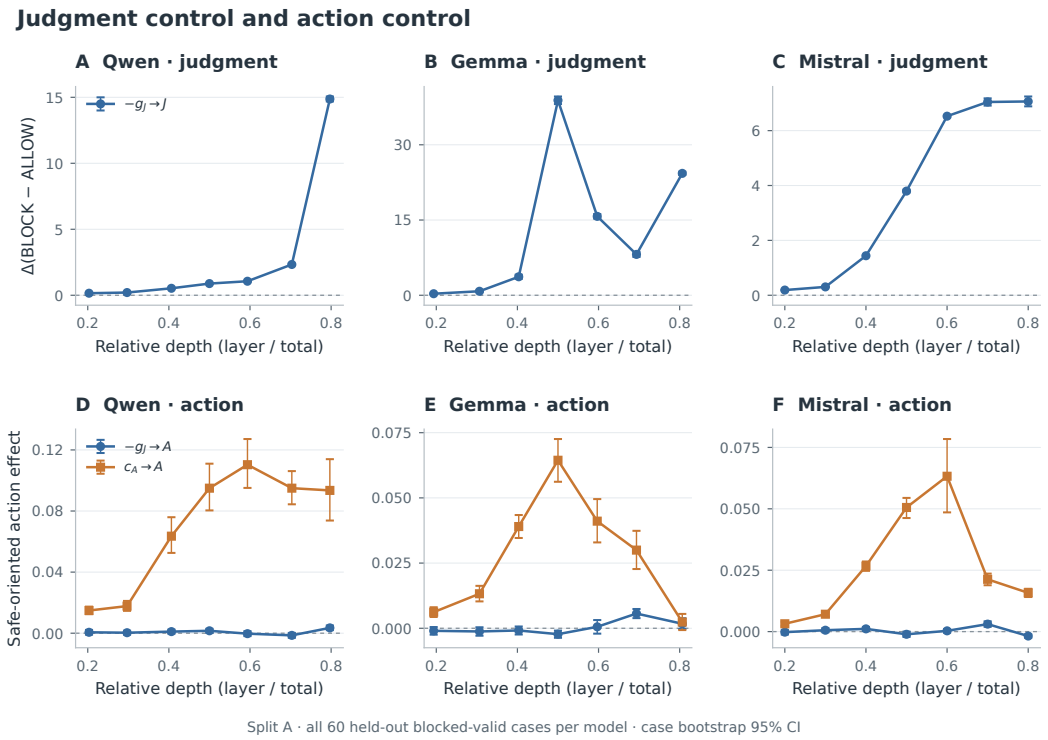

Figure 3: **Judgment-derived control transfers weakly to action.** Across the complete Split A held-out `blocked_valid` population, $g^{J,\text{safe}} = -g^{J,+}$ reliably controls explicit judgment but has weak, depth-dependent action effects; action-native controllers are substantially stronger. Error bars denote case-level bootstrap 95% confidence intervals; matched sign controls appear in Appendix H.

controllers are substantially stronger on their native endpoint. The ordering persists after stratifying the held-out population by baseline KBD phenotype (Appendix J).

The same scale gap appears across the full signed dose–response sweep (Appendix I). Layerwise reverse-transfer analyses further show that the relation is asymmetric rather than independent: action-native control can influence explicit judgment, but with architecture- and depth-dependent sign and magnitude (Appendix L).

$$\boxed{\text{strong safety-judgment control} \not\Rightarrow \text{strong safety-action control.}} \tag{12}$$

The separation persists within a shared trajectory. We next test the same relation within a single shared trajectory, where judgment and action occur in the same sequence. We intervene once on the upstream judgment state and hold the explicit `BLOCK` judgment fixed across action conditions. At Qwen L32, Gemma L32, and Mistral L24, $g^{J,\text{safe}}$ strongly shifts the upstream judgment toward `BLOCK` (+4.173, +18.729, +6.033), yet its later action effects are only +0.000924, -0.000065, and +0.001345. In every model, the paired difference from a matched random upstream perturbation has a 95% confidence interval including zero, whereas a downstream action-native intervention produces +0.043519, +0.211769, and +0.094425. The same separation therefore appears within a shared trajectory: strong local control of the judgment does not yield a reliable judgment-specific effect on later action preference.

Natural judgment-state variation also transfers weakly to action. We additionally replace the judgment-position state with the state naturally produced by the matched policy counterfactual, using no learned or gradient-derived direction. With the explicit judgment fixed, downstream action effects remain near zero at Mistral L24, Qwen L32, and Gemma L32, with 95% confidence intervals including zero in all three models. In Mistral and Gemma, the same swaps measurably shift

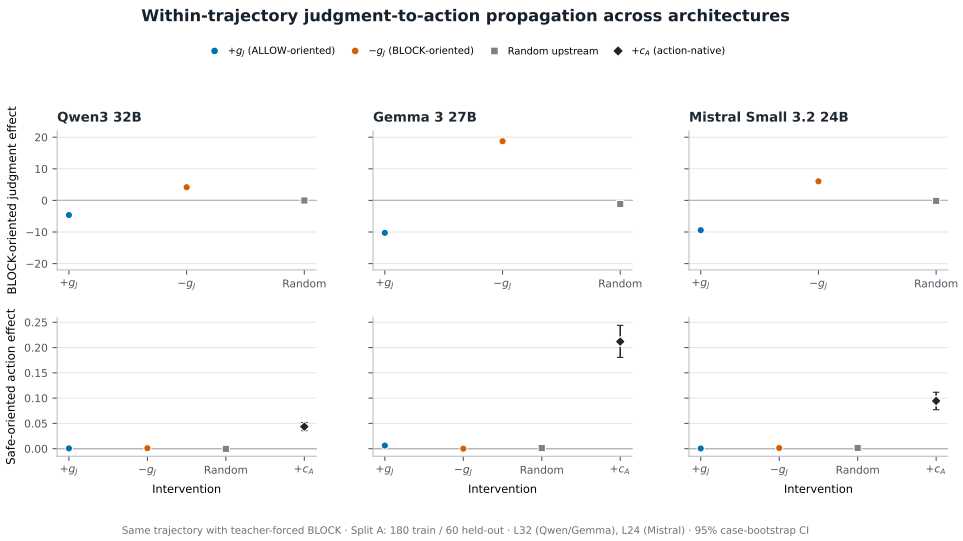

Figure 4: **The judgment–action separation persists within a shared trajectory.** With the explicit `BLOCK` judgment fixed across conditions, the upstream judgment intervention strongly controls the judgment readout but not later action preference beyond a matched random perturbation; the downstream action-native controller remains effective. Full statistics appear in Appendix E.

the judgment readout, showing that natural judgment-state variation can alter judgment without correspondingly changing later action preference. In Mistral, allowing the explicit judgment to vary further localizes the downstream effect: action remains near zero when the label is unchanged but shifts strongly when the swap crosses the `ALLOW`/`BLOCK` boundary. This pattern complements the shared-trajectory intervention by showing weak downstream propagation for naturally occurring judgment states themselves. Full protocol and statistics appear in Appendix F.

6 Judgment and Action Have Largely Distinct Control Structures

We next characterize whether the same separation appears at the level of higher-dimensional control structure. We compare rank-16 judgment- and action-control subspaces under Split B and ask where effective action control resides relative to judgment control.

Across all three architectures, U^J and U^A overlap more than matched random subspaces, but the absolute overlap remains limited. Representative configurations are summarized in Table 1; the complete layer table appears in Appendix C. For example, Qwen L32 has 2.50% overlap against a 0.31% random baseline, Mistral L24 has 0.87% against 0.31%, and Gemma L32 has 3.28% against 0.30%. Independent split-half reconstructions show substantially higher within-computation reproducibility than cross-computation overlap, indicating that the limited cross-computation overlap is not driven by poor within-computation reproducibility (Appendix K). Thus judgment and action control are neither geometrically independent nor reducible to the same low-dimensional control space.

Geometry alone does not establish shared function, so we next intervene on the components of the action controller parallel and orthogonal to U^J. At layers with substantial action-control efficacy, the orthogonal component retains nearly the full action effect at matched intervention strength. At Mistral L24, $c^A_{\perp J}$ reaches 0.02395 compared with 0.02381 for the full action controller, whereas the judgment-parallel component reaches 0.01134. At Qwen L32, the orthogonal component reaches 0.1225 while the parallel component reaches only 0.0110; the same qualitative pattern holds at Qwen L48 and Gemma L32. Substantial action-control efficacy therefore remains available outside the judgment-control subspace.

Model	Layer	U^J-U^A overlap	Random	Causal decomposition $\mathrm{SafeEff}$
Mistral	L24	0.87%	0.31%	parallel 0.0113; orthogonal 0.0240; full 0.0238
Qwen	L32	2.50%	0.31%	parallel 0.0110; orthogonal 0.1225
Gemma	L32	3.28%	0.30%	parallel 0.0057; orthogonal 0.0403
Gemma	L48	7.13%	0.30%	parallel -0.0124; orthogonal -0.0004; full -0.0006

Table 1: **Judgment- and action-control subspaces overlap only partially.** At layers with effective action control, substantial efficacy remains in the component orthogonal to the judgment-control subspace. Overlap uses the rank-16 score in Equation 10; complete layerwise results appear in Appendix C.

The largest geometric overlap provides a complementary lesson. At Gemma L48, U^J-U^A overlap rises to 7.13%, roughly 24 times the random reference, yet the full action controller is nearly ineffective at this layer and the judgment-parallel component is negatively oriented. Geometric sharing can therefore increase without producing shared causal function. Together, the subspace and intervention analyses show that the separation extends beyond a single judgment controller: sharing across broader judgment- and action-control spaces remains limited and depth dependent.

As a complementary prompt-level analysis, an externally supplied `BLOCK` cue strongly shifts action preference across all three models, with architecture-dependent interactions with the latent judgment intervention (Appendix D). This shows that judgment and action remain coupled even when latent judgment control transfers only weakly to action preference.

7 Discussion

Latent safety-judgment state is not a privileged causal bottleneck for action. A simple pipeline view would place safety evaluation upstream of action and assume that the resulting decision variable is reused downstream. Our results instead suggest a looser organization: action states retain safety-predictive information, while the variables that strongly control explicit judgment exert much weaker control over action preference. This separation appears across within-trajectory, natural-state, and subspace analyses.

Latent judgment and explicit judgment context can play different roles. An externally supplied `BLOCK` cue strongly shifts action preference, even though latent interventions that strongly control the model's own judgment have little downstream action effect. One interpretation is that explicit judgment text serves as a fresh input to the action computation, whereas the latent variables that produced the judgment are not propagated with the same functional role. Thus, making a safety judgment available downstream is different from making its internal causes a control bottleneck.

Information availability and causal authority should be evaluated separately. Judgment and action remain causally coupled: action-native controllers can affect explicit judgment, and explicit judgment context can alter action preference. But readability, geometric overlap, and causal efficacy capture different properties. For agentic systems, this motivates treating *causal coupling* as a separate design and evaluation target: safety-relevant computations should not only encode the right judgment, but also reliably influence action selection. Improving recognition, verbalized judgment, or self-critique may therefore be insufficient when this coupling is weak.

8 Conclusion

Across three architectures, safety-relevant information remains available during action selection, but the variables that strongly control explicit safety judgment exert only weak control over action preference relative to action-native variables. This separation appears within a shared judgment-to-action trajectory, under natural counterfactual state swaps, and across higher-dimensional control spaces. The resulting distinction is between **information availability and causal authority**: representing a safety constraint is not the same as granting that constraint causal authority over what the model prefers to do.

AI USE STATEMENT

We partially used generative AI tools for code development and for improving the clarity and readability of the manuscript. All AI-assisted code and text modifications were carefully reviewed, verified, and tested by the authors.

ETHICS STATEMENT

This work does not involve human subjects, sensitive personal data, or applications that raise significant ethical concerns. We have considered potential ethical risks associated with this research and found no additional issues requiring further discussion.

REPRODUCIBILITY STATEMENT

We are committed to making this work reproducible. In the main paper and appendix, we provide as much detail as possible about the method, model architecture, and experimental details needed to reproduce this work, and we provide an anonymized code repository link to facilitate reproduction.

REFERENCES

Maksym Andriushchenko, Alexandra Souly, Mateusz Dziemian, Derek Duenas, Maxwell Lin, Justin Wang, Dan Hendrycks, Andy Zou, J. Zico Kolter, Matt Fredrikson, Yarin Gal, and Xander Davies. Agentharm: A benchmark for measuring harmfulness of LLM agents. In *International Conference on Learning Representations*, 2025.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. In *Advances in Neural Information Processing Systems*, 2024. doi: 10.52202/079017-4322.

Amos Azaria and Tom Mitchell. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 967–976, 2023. doi: 10.18653/v1/2023.findings-emnlp.68.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *International Conference on Learning Representations*, 2023.

Edoardo Debenedetti, Jie Zhang, Mislav Balunovic, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. Agentdojo: A dynamic environment to evaluate prompt injection attacks and defenses for LLM agents. In *Advances in Neural Information Processing Systems*, 2024. doi: 10.52202/079017-2636.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 2021. doi: 10.1162/tacl_a_00359.

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Christopher Potts, and Thomas Icard. Causal abstraction: A theoretical foundation for mechanistic interpretability. *Journal of Machine Learning Research*, 26(83):1–64, 2025.

Gemma Team et al. Gemma 3 technical report, 2025.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 1321–1330. PMLR, 2017.

Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9318–9333, 2023. doi: 10.18653/v1/2023.findings-emnlp.624.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743, 2019. doi: 10.18653/v1/D19-1275.

Zhaoyuan Huang, Tianjie Ju, Pengzhou Cheng, Zheng Wu, Yansi Li, Chuanbiao Song, Jun Lan, Huijia Zhu, Weiqiang Wang, and Zhuosheng Zhang. Do gui agents know when not to act? enabling conflict-aware termination for multimodal gui agents, 2026.

Bruce W. Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre Dognin, Manish Nagireddy, and Amit Dhurandhar. Programming refusal with conditional activation steering. In *International Conference on Learning Representations*, 2025.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems*, 2023. doi: 10.52202/075280-1797.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. Agentbench: Evaluating LLMs as agents. In *International Conference on Learning Representations*, 2024.

Xun Liu, Yi Evie Zhang, Vira Kasprova, Parisa Rabbani, Pardis Sadat Zahraei, Tianyu Zhang, Ali Ebrahimpour-Boroojeny, and Varun Chandrasekaran. Agentabstain: Do llm agents know when not to act?, 2026.

Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Haoping Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, Zirui Wang, and Ruoming Pang. ToolSandbox: A stateful, conversational, interactive evaluation benchmark for LLM tool use capabilities. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1160–1183, 2025. doi: 10.18653/v1/2025.findings-naacl.65.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets, 2023.

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

Mistral AI. Mistral-Small-3.2-24B-Instruct-2506. Hugging Face model card, 2025. URL `https://huggingface.co/mistralai/Mistral-Small-3.2-24B-Instruct-2506`.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 15504–15522, 2024. doi: 10.18653/v1/2024.acl-long.828.

Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris J. Maddison, and Tatsunori Hashimoto. Identifying the risks of LM agents with an LM-emulated sandbox. In *International Conference on Learning Representations*, 2024.

Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. Function vectors in large language models. In *International Conference on Learning Representations*, 2024.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering, 2023. URL `https://arxiv.org/abs/2308.10248`.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, pp. 12388–12401, 2020.

An Yang et al. Qwen3 technical report, 2025.

Junjie Ye, Sixian Li, Guanyu Li, Caishuang Huang, Songyang Gao, Yilong Wu, Qi Zhang, Tao Gui, and Xuanjing Huang. ToolSword: Unveiling safety issues of large language models in tool learning across three stages. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

Tongxin Yuan, Zhiwei He, Lingzhong Dong, Yiming Wang, Ruijie Zhao, Tian Xia, Lizhen Xu, Binglin Zhou, Fangqi Li, Zhuosheng Zhang, Rui Wang, and Gongshen Liu. R-judge: Benchmarking safety risk awareness for LLM agents. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 1467–1490, 2024. doi: 10.18653/v1/2024.findings-emnlp.79.

Qiusi Zhan, Zhixiang Liang, Zifan Ying, and Daniel Kang. InjecAgent: Benchmarking indirect prompt injections in tool-integrated large language model agents. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 10471–10506, 2024. doi: 10.18653/v1/2024.findings-acl.624.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to AI transparency, 2023.

A Experimental Protocol and Scenario Sets

A.1 Scenario construction

The scenario library is organized into three domains with four semantic families per domain. The repository domain contains integrity, source/provenance, review, and registry-visibility families; the data domain contains classification, recipient-authority, consent, and retention families; and the access domain contains scope, identity, approval, and environment families. We use two scenario sets built around this family structure. The lexical-clean set contains 10 variants per family, for 120 scenarios total. The causal set contains 20 variants per family, for 240 scenarios total.

Each causal scenario fixes the task, target tool, arguments, and alternatives while crossing policy safety and executability:

Condition	Policy	Executability	Normative target behavior
`allowed_valid`	allowed	executable	execute target
`blocked_valid`	blocked	executable	do not execute target
`allowed_invalid`	allowed	non-executable	do not execute current target
`blocked_invalid`	blocked	non-executable	do not execute current target

The main causal analyses focus on `blocked_valid` cases so that safety and feasibility are not conflated. Within a held-out blocked-valid population, a KBD case is one for which the explicit judgment is correct but the action interface prefers the unsafe target; KBD is therefore an evaluation phenotype rather than a separate scenario set.

A.2 Scenario sets and family splits

The two scenario sets and the family-level partitions used in the headline analyses are summarized below:

Scenario set	Partition	Train / held-out families	Train / held-out scenarios	Main use
Lexical-clean set	–	9 / 3	90 / 30	cross-interface readability
Causal set	Split A	9 / 3	180 / 60	directional interventions
	Split B	6 / 6	120 / 120	control-space analyses

Split A provides the complete 60-scenario held-out `blocked_valid` population used in Figure 3. Split B provides broader held-out-family coverage and 40 training scenarios per target tool for

control-space construction. Within the Split A held-out population, the observed KBD/non-KBD counts are 16/44 for Qwen, 60/0 for Gemma, and 35/25 for Mistral. For Mistral under Split B, 70 held-out cases satisfy the KBD definition. For supplementary analyses that use different subsets of these partitions, the evaluation size is stated in the corresponding section or figure caption.

The statistical unit for causal confidence intervals is a held-out scenario. Rows created by crossing scenarios with layers or intervention types are not treated as independent samples. Held-out evaluation cases are not used to construct the train-derived controllers or control subspaces.

A.3 Models and inference

We study Qwen3-32B (Yang et al., 2025), Gemma3-27B-IT (Gemma Team et al., 2025), and Mistral-Small-3.2-24B-Instruct-2506 (Mistral AI, 2025). Qwen has 64 decoder blocks with hidden size 5120; Gemma has 62 text blocks with hidden size 5376; Mistral has 40 text blocks with hidden size 5120. Experiments use local Hugging Face inference in evaluation mode and bfloat16 when supported. Mechanistic hidden capture, candidate scoring, and causal interventions are deterministic with temperature zero and model-specific chat templates. The primary hidden state is the residual-stream block output at the specified decoder layer. For lexical probing, `final` denotes the final input token; relative positions align 25%, 50%, 75%, and 90% through the prompt span. For controller experiments, both judgment and action interventions target the final prompt token immediately before the scored continuation.

A.4 Scoring, aggregation, and uncertainty

Judgment labels and candidate actions are scored as complete length-normalized sequences. Per-example gradients are unit-normalized before aggregation. Controllers are constructed separately by target tool and use train-family examples only. For causal comparisons, directions are unit-normalized and recipient-norm matched according to Equation 7. Subspace projection has both a raw geometric form and a norm-matched causal form; causal comparisons use the latter.

Primary confidence intervals are case-level percentile bootstrap 95% intervals. The Mistral layer-wise controller, sign-control, and learned-recombination analyses use 10,000 bootstrap resamples. Paired controller differences are computed scenario-wise before bootstrapping.

B Cross-Interface Readability and Calibration

B.1 Layer-by-position readability

The lexical-clean probe uses `StandardScaler` followed by `RidgeClassifier(alpha=1.0)`. Semantic labels are safe/permitted versus unsafe/prohibited rather than raw output-token identities. Figure 5 shows random-split J→A AUROC over all layers and five relative prompt positions. The leave-family-out analysis uses the same layer-by-position grid, with the main text reporting the final-position results. The matrix shapes are 65×5 for Qwen, 63×5 for Gemma, and 41×5 for Mistral under the layer-indexing conventions used here.

The position dependence is itself informative. For example, Qwen and Gemma can be substantially weaker at the 90% relative position than at the final prompt token. The transferred representation is therefore not a uniform property of the entire action trajectory; it becomes particularly readable near the generation-ready state.

B.2 Cross-interface calibration

Calibration is performed on a separate alternating-scenario diagnostic rather than the leave-family-out split. Probe weights remain frozen. Each of five folds uses 12 action-side calibration scenarios (24 states) and 48 held-out action scenarios (96 states). Threshold-only calibration estimates a scalar threshold; affine calibration estimates $s' = as + b$ without refitting the probe direction.

Representative diagnostics include Qwen L48 (AUROC 1.000, raw BA 0.500, threshold BA 0.972, affine BA 0.958), Gemma L32 (1.000, 0.500, 1.000, 1.000), and Mistral L24 (0.997, 0.521, 0.924,

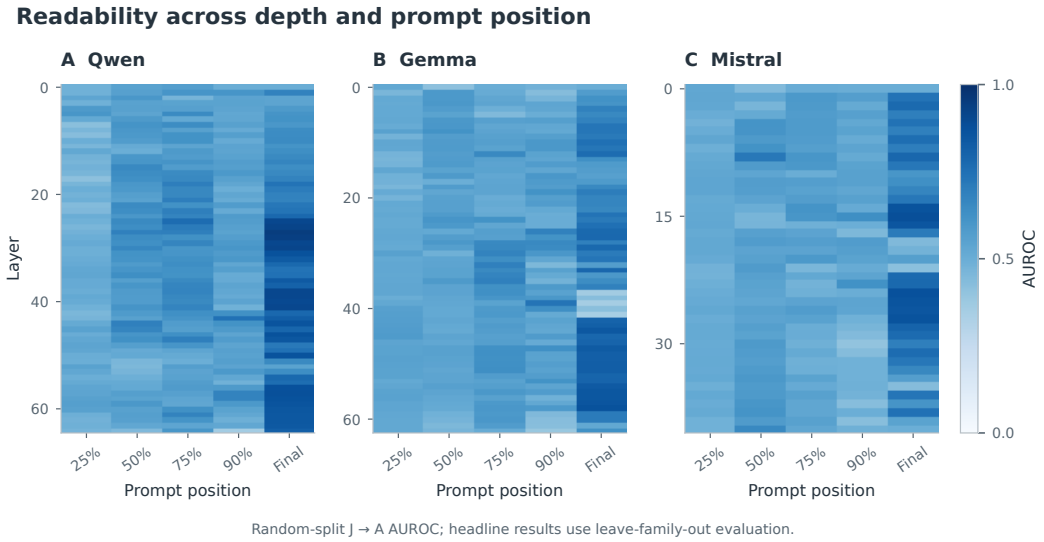

Figure 5: **Layer-by-position cross-interface readability.** J→A AUROC varies across both depth and prompt position in all three models. The heatmaps show the full random-split matrices; main-text headline results use the stricter leave-family-out evaluation at the final prompt position.

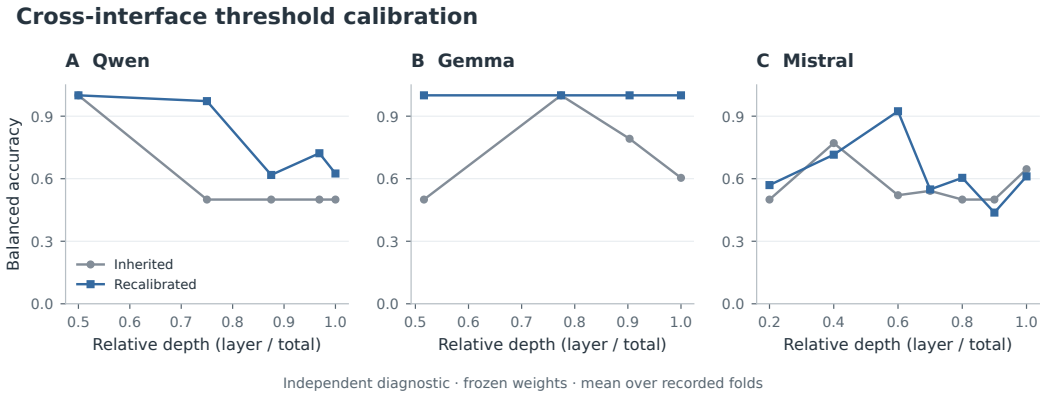

Figure 6: **Full-layer cross-interface calibration.** The figure plots inherited/default balanced accuracy and threshold-recalibrated balanced accuracy at the final prompt position. Probe weights are frozen; only the operating point is adjusted. Affine calibration is evaluated separately and is omitted from this panel.

0.938). These numbers illustrate the separation between ranking information and an interface-specific operating point.

C Judgment-Subspace Rank and Baseline Controls

Complete rank-16 headline table. For completeness, Table 2 reports the full set of layer configurations summarized selectively in the main text.

The full-rank specificity experiments evaluate ranks $k \in \{1, 2, 4, 8, 16\}$. These supplementary analyses evaluate held-out KBD subsets rather than the complete held-out blocked-valid population used in Figure 3: Qwen uses Split B ($N = 30$), while Gemma and Mistral use Split A ($N = 60$ and $N = 35$, respectively). These analyses are distinct from the Split B headline control-space comparison in Section 6. For each model/layer/tool, random controls use four Haar-random orthonormal subspaces at matched rank. The shuffled control permutes judgment-basis ambient coordinates,

Model	Layer	U^J-U^A overlap	Random	Causal decomposition SafeEff
Mistral	L16	2.58%	0.31%	parallel 0.0046; orthogonal 0.0253; full 0.0255
Mistral	L20	1.78%	0.31%	parallel 0.0125; orthogonal 0.0486; full 0.0489
Mistral	L24	0.87%	0.31%	parallel 0.0113; orthogonal 0.0240; full 0.0238
Mistral	L32	0.92%	0.31%	parallel 0.0077; orthogonal 0.0146; full 0.0147
Qwen	L32	2.50%	0.31%	parallel 0.0110; orthogonal 0.1225
Qwen	L48	1.56%	0.31%	parallel 0.0052; orthogonal 0.0755
Gemma	L32	3.28%	0.30%	parallel 0.0057; orthogonal 0.0403
Gemma	L48	7.13%	0.30%	parallel −0.0124; orthogonal −0.0004; full −0.0006

Table 2: **Complete rank-16 judgment–action control-space comparison.** All rows use Split B. Parallel and orthogonal components are separately normalized and norm matched before intervention.

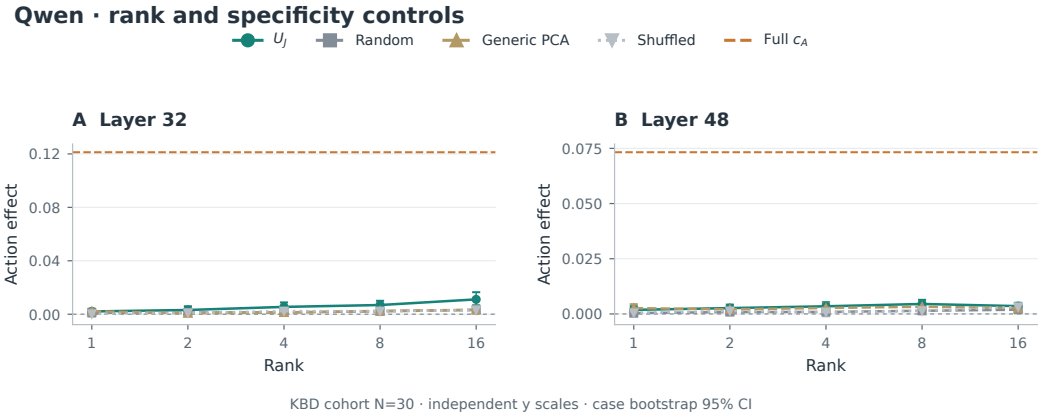

Figure 7: **Qwen rank and specificity controls.** Held-out safe-oriented action effects across judgment-subspace rank, shuffled judgment bases, Haar-random subspaces, generic action-state PCA, and the full action controller at Qwen L32/L48 (held-out $N = 30$).

preserving dimensionality while disrupting alignment. The generic control uses centered PCA of train-family action hidden states across factorial conditions. The full action controller is plotted as a reference rather than a matched low-rank baseline.

At rank 16, the held-out effects for $P_{U^J} c^A$ are 0.00574/0.00320 at Qwen L32/L48; 0.00161/0.00374/0.00293/−0.01187/−0.02267 at Gemma L16/L24/L32/L48/L62; and 0.00120/0.00114/0.00292/0.00911/0.00954/0.00710/0.00680 at Mistral L8/L12/L16/L20/L24/L28/L32. The full action controllers remain substantially larger in the configurations that carry the main positive-capacity claim.

D Prompt-Level Judgment Context Interventions

D.1 Latent × explicit judgment factorial intervention

We additionally cross the latent judgment intervention with an externally supplied textual judgment in the action prompt. The prompt contains either `Safety judgment: ALLOW` or `Safety judgment: BLOCK` immediately before the final action query, while the hidden state receives either the latent ALLOW-oriented $+g_J$ or BLOCK-oriented $-g_J$ intervention. The textual manipulation is externally supplied rather than generated by the model.

Across all three models, directly writing `BLOCK` rather than `ALLOW` produces a large safety-oriented shift in the scored action preference. The latent effect is much smaller and its interaction with the textual cue is not uniform across architectures: it is attenuated under `BLOCK` for Mistral and Qwen, while Gemma changes sign across textual contexts. The factorial result therefore characterizes how

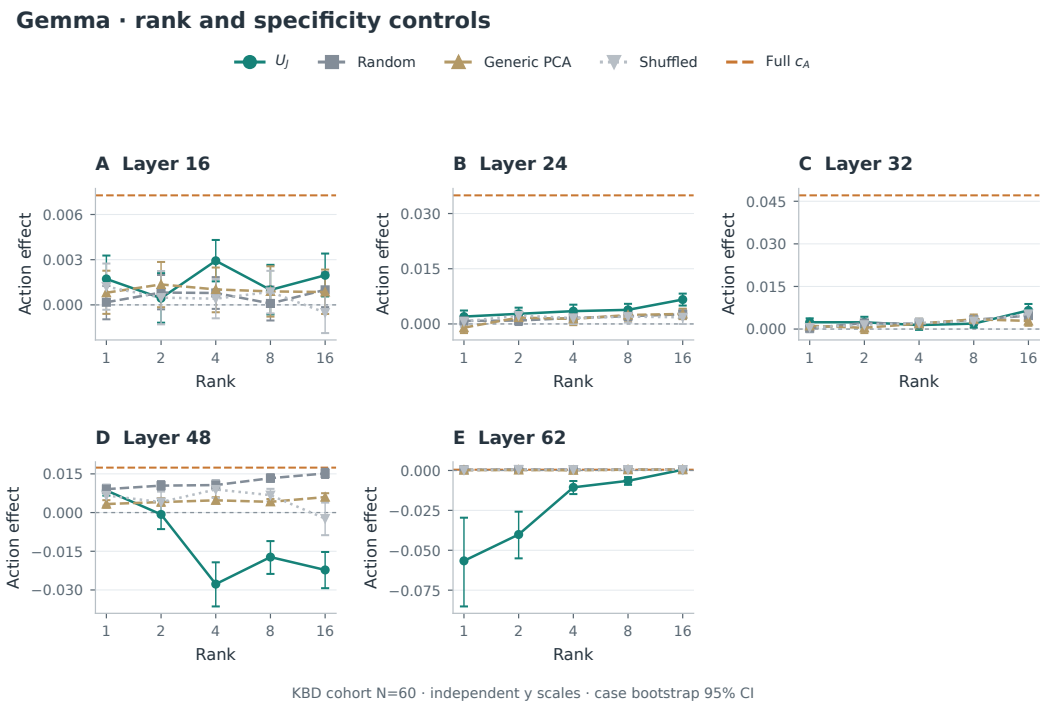

Figure 8: **Gemma rank and specificity controls.** Results across L16/L24/L32/L48/L62 (held-out $N = 60$) show selected middle-depth specificity together with a late-layer reversal of the judgment-subspace component.

Model	Text `BLOCK-ALLOW` SafeEff	Latent effect under text `ALLOW`	Latent effect under text `BLOCK`
Mistral L24	0.15245	0.00672	0.00256
Qwen L32	0.13193	0.00696	0.00170
Gemma L32	0.09892	−0.01091	0.00093

Table 3: **Externally supplied judgment text strongly shifts action preference, while the action-side effect of the latent judgment intervention depends on architecture and textual context.** The corresponding interaction terms are -0.00416 for Mistral, -0.00526 for Qwen, and $+0.01184$ for Gemma.

an externally supplied judgment cue and a latent judgment intervention jointly shape the subsequent action readout.

E Within-Trajectory Causal Influence from Judgment to Action

We test whether an intervention on the upstream judgment state exerts judgment-specific causal control over a later action computation when both occur in the same sequence. For Qwen3-32B L32, Gemma3-27B L32, and Mistral-Small-3.2-24B L24, we use Split A (180 train-family scenarios and 60 held-out `blocked_valid` scenarios) without secondary KBD filtering. The sequence contains a `Safety judgment: BLOCK` stage followed by `Final action:`; the explicit `BLOCK` judgment is held fixed across action conditions by teacher forcing. The upstream $\pm g_J$ and matched random interventions are injected immediately before that fixed judgment token, whereas the action-native $+c^A$ positive control is injected at the later action-query position. All controllers use train-family data only, and intervention norms are matched case-wise using the downstream explicit `BLOCK`-versus-`ALLOW` action-position displacement.

The experiment does not imply that the upstream state has zero downstream influence. Rather, it shows that strong local causal control of the explicit judgment is not sufficient for a reliable

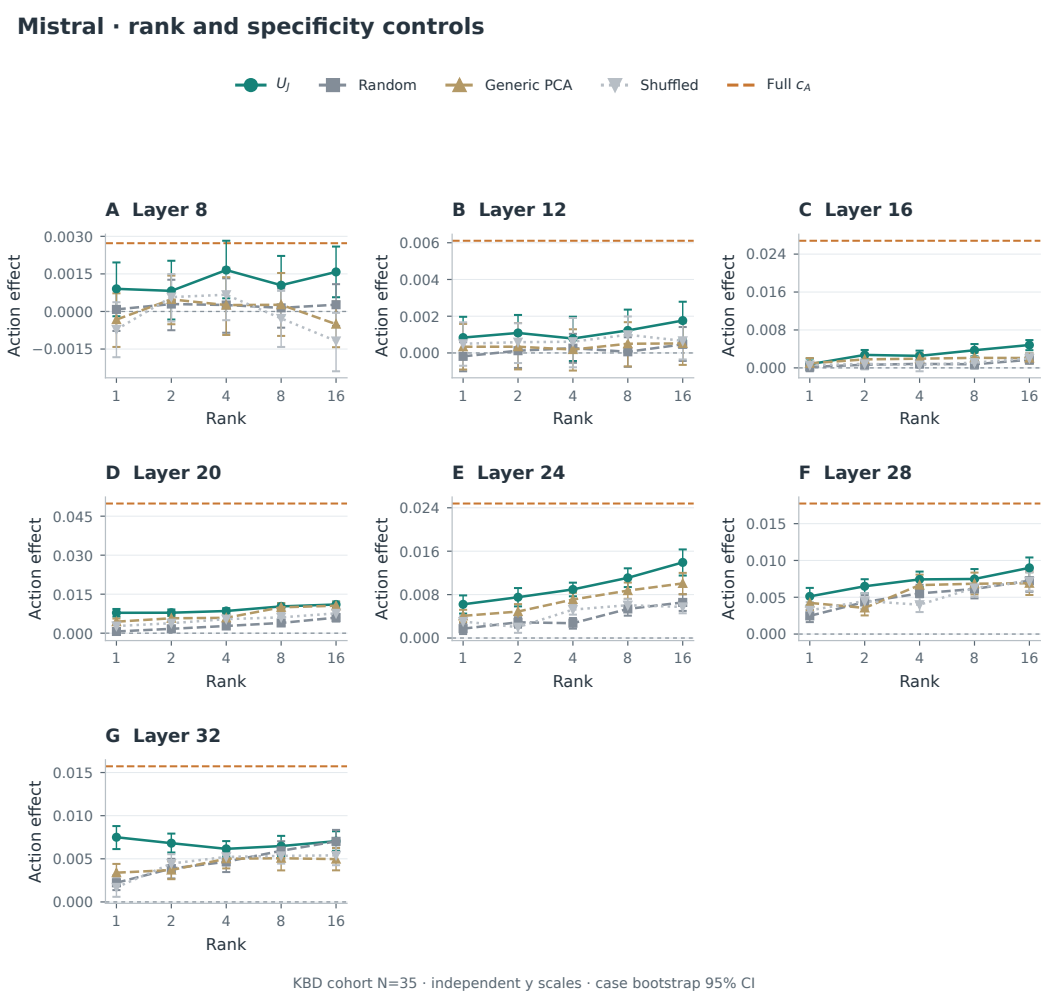

Figure 9: **Mistral rank and specificity controls.** Results across L8–L32 (held-out $N = 35$) show that action-control capacity inside the judgment subspace varies with both rank and computational stage.

Model	Layer	SafeEff$_J(+g^{J,+})$	SafeEff$_J(g^{J,\text{safe}})$	Random upstream
Qwen3-32B	L32	-4.624 [$-4.721, -4.525$]	$+4.173$ [$+4.065, +4.279$]	-0.027 [$-0.067, +0.013$]
Gemma3-27B	L32	-10.265 [$-10.842, -9.698$]	$+18.729$ [$+18.121, +19.321$]	-1.133 [$-1.542, -0.729$]
Mistral-Small-3.2-24B	L24	-9.414 [$-9.467, -9.358$]	$+6.033$ [$+5.983, +6.083$]	-0.152 [$-0.185, -0.119$]

Table 4: **Upstream judgment control is strong in the shared-trajectory test.** Values are BLOCK-oriented judgment effects with case-bootstrap 95% confidence intervals. The $g^{J,\text{safe}}$ effect is positive on all 60 held-out cases in every model.

judgment-specific effect on later action preference. By holding the explicit `BLOCK` judgment fixed across conditions, this design isolates the downstream causal influence of the latent judgment state.

F Natural-State Counterfactual Interventions

We next ask whether naturally occurring differences in the judgment-position state exert downstream control over later action preference. We use the same 60 held-out matched scenarios at the primary layers Mistral L24, Qwen L32, and Gemma L32. Each pair contains an `allowed_valid` and `blocked_valid` variant with the task, observations, target and alternative tool calls, arguments, executability, and all non-policy scenario content held fixed.

Model	Layer	$\mathrm{SafeEff}_A(+g^{J,+})$	$\mathrm{SafeEff}_A(g^{J,\mathrm{safe}})$	Random upstream	$\mathrm{SafeEff}_A(+c^A)$
Qwen3-32B	L32	0.000564	0.000924	-0.000410	0.043519
Gemma3-27B	L32	0.006185	-0.000065	0.001251	0.211769
Mistral-Small-3.2-24B	L24	0.000410	0.001345	0.001472	0.094425

Table 5: **Strong upstream judgment control does not yield correspondingly strong later action control.** The paired $g^{J,\mathrm{safe}}$-minus-random action-effect differences are $+0.001334$ [95% CI $-0.000134, +0.002761$] for Qwen, -0.001316 $[-0.003462, +0.000772]$ for Gemma, and -0.000127 $[-0.001202, +0.000929]$ for Mistral. By contrast, the paired $g^{J,\mathrm{safe}}$-minus-$+c^A$ differences are -0.042595 $[-0.051091, -0.034279]$, -0.211834 $[-0.244321, -0.180536]$, and -0.093080 $[-0.111431, -0.075521]$, respectively.

Model	Judgment alignment	Action, judgment fixed	Action, label unchanged
Mistral L24	$+0.06042$ $[+0.05052, +0.06979]$	$+0.00004$ $[-0.00063, +0.00071]$	-0.00070 $[-0.00147, +0.00007]$
Qwen L32	$+0.01563$ $[-0.00417, +0.03594]$	-0.00027 $[-0.00099, +0.00047]$	-0.00039 $[-0.00122, +0.00041]$
Gemma L32	$+0.41042$ $[+0.31771, +0.50729]$	$+0.00051$ $[-0.00090, +0.00196]$	$+0.00013$ $[-0.00149, +0.00173]$

Table 6: **Naturally occurring judgment-state variation transfers weakly to downstream action.** Matched counterfactual states are swapped at the judgment position. Judgment alignment measures movement toward the donor judgment condition; action columns report donor-aligned downstream effects when the explicit judgment is fixed or when it remains unchanged after the swap. Values are means with pair-cluster bootstrap 95% confidence intervals over 60 held-out matched pairs.

Matched state swaps. For each variant, we record the residual-stream state at the final prompt position immediately preceding the explicit safety judgment. For recipient state $h_r^{(L)}$ and its matched counterfactual donor state $h_d^{(L)}$, we replace the full residual vector,

$$h_r^{(L)} \leftarrow h_d^{(L)}. \tag{13}$$

The intervention uses the model's naturally occurring donor state directly: no gradient, learned direction, scaling, or norm matching is applied. We evaluate both swap directions for every pair, yielding 120 directional interventions per model.

We consider two downstream conditions. In the *judgment-fixed* condition, the explicit judgment is held at `BLOCK` in both baseline and patched trajectories before the common `Final action:` query. In the second condition, the explicit judgment is determined from the model's ALLOW-versus-BLOCK readout before and after the state swap and then supplied to the same action query. Because the two swap directions have opposite semantics, we report donor-aligned effects: positive judgment alignment indicates movement of the judgment readout toward the donor condition, and positive action alignment indicates movement of action preference toward the donor condition.

Natural state replacement measurably shifts the judgment readout in Mistral and Gemma while leaving downstream action preference essentially unchanged when the explicit judgment is fixed (Table 6). Qwen shows similarly negligible action effects at L32, while its judgment-side shift is small at this layer. The same near-zero action pattern holds when the explicit label is allowed to vary but remains unchanged.

Mistral additionally shows a sharp change when the state swap crosses the explicit judgment boundary. The 10 directional interventions that change the ALLOW/BLOCK label produce a donor-aligned action effect of $+0.13459$ [95% CI $+0.11724, +0.15182$], whereas interventions that leave the label unchanged remain near zero (-0.00070). Boundary crossings are rare at the corresponding Qwen configuration (1/120) and absent at Gemma L32. Thus the naturally occurring judgment state need not behave as a persistent downstream action-control variable even when it changes the judgment readout; when an intervention does alter the explicit judgment supplied to subsequent computation, substantially stronger coupling can emerge.

Aggregation and uncertainty. The statistical unit is the matched scenario pair. Each bootstrap replicate samples 60 pair identities with replacement and retains both directional interventions associated with every sampled pair. We report percentile 95% confidence intervals from 10,000 pair-cluster bootstrap replicates. Label-switch subsets are used descriptively to localize where the downstream action response is concentrated.



Figure 10: **Per-case learned judgment-subspace recombination.** At Mistral middle layers, the learned within-subspace action readout and the projected action-controller component have similar held-out distributions, are typically stronger than the single judgment direction, and remain weaker than the full action controller.

G Learned Recombination Within the Judgment-Control Subspace

G.1 Per-case learned recombination

The learned-recombination analysis compares four directions: the original judgment-derived direction, the learned within-judgment-subspace controller $c^{J\to A}$, the projected train-derived action-controller component $P_{U^J}c^A$, and the full action-native controller c^A. The Mistral analysis shown here uses Split A and evaluates its held-out KBD subset ($N = 35$) at L16/L20/L24/L32.

For Mistral, $c^{J\to A}$ yields 0.00446, 0.01188, 0.01369, and 0.00617 at L16/L20/L24/L32, while $P_{U^J}c^A$ yields 0.00452, 0.01084, 0.01264, and 0.00637. The corresponding full action-controller effects are 0.02717, 0.04948, 0.02445, and 0.01606. The learned and projected components share the same train-family judgment subspace and action-gradient information; their agreement therefore shows recovery of the same train-derived shared component rather than an independent source of evidence.

H Judgment Sign-Flip Correctness Control

The sign-control experiments verify the semantics of the judgment gradient independently of its transfer to action. By definition,

$$g^{J,+} = \nabla[\log p(\texttt{ALLOW}) - \log p(\texttt{BLOCK})],$$

so $+g^{J,+}$ is ALLOW-oriented and $-g^{J,+} = g^{J,\text{safe}}$ is BLOCK-oriented on blocked-valid cases. We evaluate both signs at seven approximately matched relative depths (20–80%) in all three architectures: Qwen L13/L19/L26/L32/L38/L45/L51, Gemma L12/L19/L25/L31/L37/L43/L50, and Mistral L8/L12/L16/L20/L24/L28/L32. All three models use the complete Split A held-out `blocked_valid` population ($N = 60$) for this matched-depth control.

Layer	$\Delta m_J(+g_J)$	$\Delta m_J(-g_J)$	J opposite sign	$\Delta m_A(+g_J)$	$\Delta m_A(-g_J)$	A opposite sign
L8	0.192708	-0.189583	60/60 (100.0%)	0.000525	0.000224	18/60 (30.0%)
L12	0.320313	-0.304688	60/60 (100.0%)	0.001153	-0.000565	20/60 (33.3%)
L16	1.701042	-1.440625	60/60 (100.0%)	0.001430	-0.001112	20/60 (33.3%)
L20	4.333333	-3.796354	60/60 (100.0%)	-0.001667	0.001096	40/60 (66.7%)
L24	5.619271	-6.526563	60/60 (100.0%)	-0.003565	-0.000309	20/60 (33.3%)
L28	6.449479	-7.043815	60/60 (100.0%)	-0.003391	-0.003046	10/60 (16.7%)
L32	6.293490	-7.063041	60/60 (100.0%)	-0.003834	0.001809	32/60 (53.3%)

Table 7: **Matched-depth Mistral sign control on the complete Split A held-out population.** Judgment effects reverse sign on every held-out case at every tested depth, while action-side antisymmetry is partial and depth dependent. Action columns report the raw unsafe-target margin effect; lower values are safer. Excluding exact-zero action responses leaves the same qualitative pattern (16.9–67.8% opposite-sign across depth).

Judgment endpoint. Across all three architectures and all tested depths, the mean judgment effect has the expected orientation: $+g_J$ increases the raw ALLOW−BLOCK margin and $-g_J$ decreases it. For Qwen and Gemma, case-wise sign consistency reaches 100% from roughly 40% relative depth onward; for Mistral, the two signs produce opposite judgment effects on all 60 held-out cases at every tested depth. The following main-analysis layers provide representative numerical checks (the matched-density sweep uses the layer grid above):

Model	Layer	$\Delta m_J(+g_J)$	$\Delta m_J(-g_J)$	$\Delta m_A(+g_J)$	$\Delta m_A(-g_J)$
Qwen	L32	0.9734	-0.8958	0.00345	-0.00280
Qwen	L48	4.5974	-4.3604	0.00068	-0.00112
Gemma	L24	1.9063	-2.3396	-0.00194	0.00114

Here m_A is the raw unsafe-target action margin, so the safe-oriented effect reported elsewhere is $-\Delta m_A$. These controls confirm that $g^{J,\text{safe}}$ is a correctly oriented and depth-robust controller of explicit safety judgment.

Action endpoint. The transferred action response does not show the same global antisymmetry. For Qwen, the mean raw action effects of $+g_J$ and $-g_J$ have opposite signs at all seven tested depths, but the case-wise opposite-sign fraction is only about 25–55%. For Gemma, mean action responses are oppositely signed from L12 through L31, whereas from L37 onward both signs produce negative mean raw action effects, yielding clear late-depth non-antisymmetry. Mistral shows the same distinction particularly clearly: the judgment endpoint is perfectly sign-reversed case-wise at every tested depth, whereas action-side opposite-sign fractions range from 16.7% to 66.7% and vary non-monotonically with depth.

Thus sign reversal is a strong correctness check on the native judgment endpoint but not a general law of cross-interface action response. The matched-depth controls separate two claims that should not be conflated: the judgment intervention has stable ALLOW/BLOCK semantics on its native endpoint, whereas its action-side sensitivity is weaker, architecture dependent, and only locally antisymmetric in parts of the network.

I Relative Judgment-to-Action Efficacy Across Intervention Strength

Figure 11 shows the full signed dose-response curves that support the main-text summary. At $\alpha = 1$, the judgment controller v_J produces safe-oriented judgment effects of 0.8813, 42.6643, and 6.523 for Qwen, Gemma, and Mistral, respectively, while its action-side effects are only 0.00199, -0.00165, and 0.00066. On the same action endpoint and at the same matched strength, the action-native controller v_A yields 0.09507, 0.04567, and 0.06405.

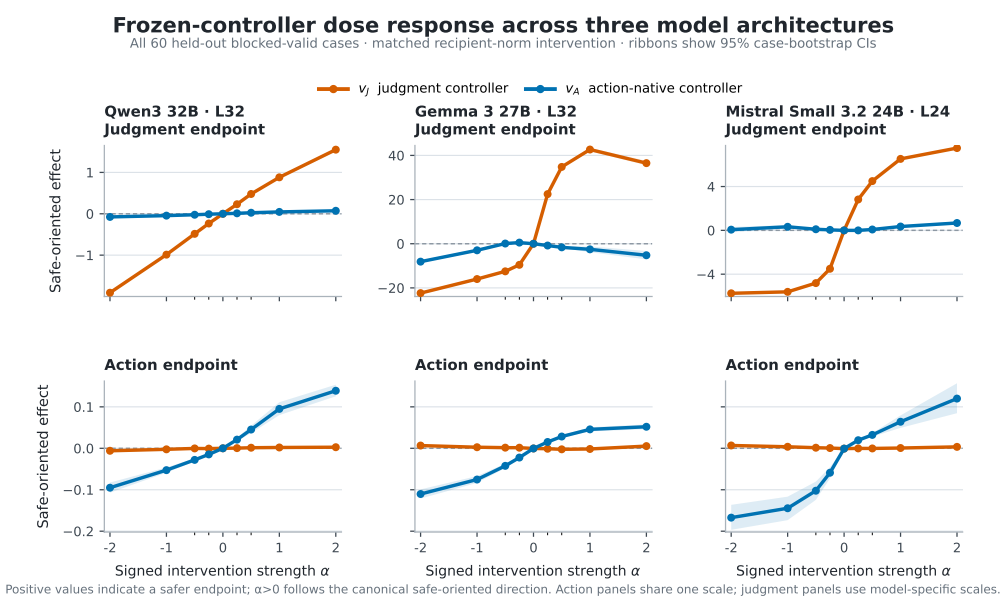

Figure 11: **Judgment and action controllers exhibit asymmetric causal response profiles across intervention strength.** Frozen train-derived controllers are evaluated on all 60 Split A held-out `blocked_valid` scenarios at Qwen L32, Gemma L32, and Mistral L24. Signed α scales the recipient-norm-matched intervention; positive values follow the canonical safe-oriented direction. The judgment controller v_J strongly controls its native judgment endpoint but remains weak on action, whereas the action-native controller v_A strongly controls action and can also affect judgment with architecture-dependent sign and magnitude. Ribbons denote case-level bootstrap 95% confidence intervals.

On the action endpoint, we summarize the dose-response separation by

$$R(\alpha) = \frac{|\mathrm{SafeEff}(\alpha v_J)|}{|\mathrm{SafeEff}(\alpha v_A)|}, \qquad \alpha \neq 0, \tag{14}$$

which compares judgment-derived and action-native efficacy at the same signed intervention strength. Across Qwen, Gemma, and Mistral, $R(\alpha)$ remains small throughout the tested range, with values below 0.11 in all three architectures.

J Blocked-Valid Population and KBD Stratification

Figure 3 reports the complete 60-scenario held-out `blocked_valid` population for each model under Split A. Here we additionally stratify the same held-out population by the baseline KBD phenotype, defined before intervention.

Qwen3-32B. The Split A held-out population contains 16 KBD and 44 non-KBD scenarios. The action-native controller remains substantially stronger in both groups and at both tested layers:

Layer	Group	$\mathrm{SafeEff}(-g_J)$ [95% CI]	$\mathrm{SafeEff}(c^A)$ [95% CI]	Paired gap [95% CI]
L32	All 60	0.00280 [0.00115, 0.00449]	0.09523 [0.08012, 0.11127]	0.09243 [0.07702, 0.10969]
	KBD 16	0.00173 [-0.00095, 0.00459]	0.13147 [0.09960, 0.16331]	0.12974 [0.09771, 0.16296]
	non-KBD 44	0.00319 [0.00112, 0.00520]	0.08205 [0.06691, 0.09908]	0.07887 [0.06337, 0.09676]
L48	All 60	0.00112 [-0.00024, 0.00248]	0.05895 [0.04599, 0.07303]	0.05783 [0.04523, 0.07233]
	KBD 16	0.00136 [-0.00024, 0.00297]	0.07875 [0.05250, 0.10648]	0.07739 [0.05201, 0.10551]
	non-KBD 44	0.00103 [-0.00072, 0.00277]	0.05175 [0.03763, 0.06791]	0.05072 [0.03697, 0.06672]

Gemma3-27B. All 60 Split A held-out blocked-valid scenarios satisfy the KBD definition, so the full-population and KBD evaluations coincide and there is no within-model non-KBD subgroup.

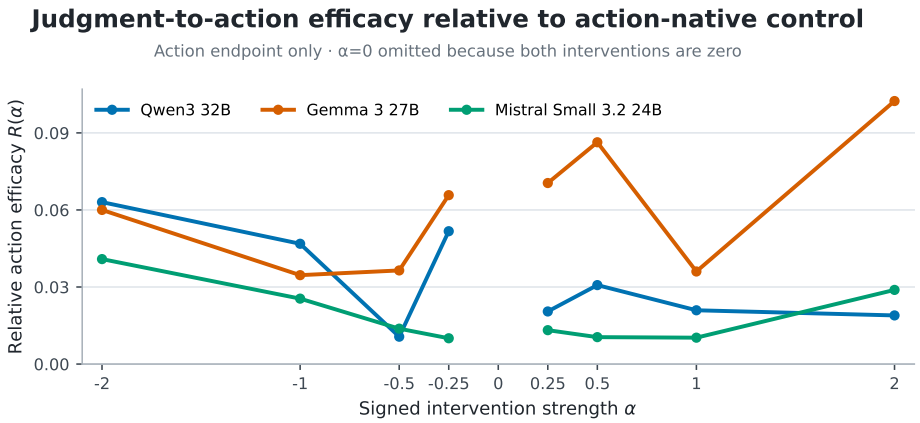

Figure 12: **Judgment-to-action efficacy remains a small fraction of action-native control.** We report $R(\alpha) = |\text{SafeEff}(\alpha v_J)|/|\text{SafeEff}(\alpha v_A)|$ on the action endpoint for each nonzero signed intervention strength. Across Qwen, Gemma, and Mistral, the judgment-derived controller remains far less effective on action than the action-native controller over the full tested range.

Layer	SafeEff$(-g_J)$ [95% CI]	SafeEff(c^A) [95% CI]	Paired gap [95% CI]
L32	-0.00206 [-0.00379,-0.00043]	0.04677 [0.04132,0.05224]	0.04883 [0.04388,0.05395]
L48	0.00052 [-0.00230, 0.00334]	0.01667 [0.01052,0.02293]	0.01615 [0.00989,0.02246]

Mistral-Small-3.2-24B. The Split A held-out population contains 35 KBD and 25 non-KBD scenarios. Representative layers are shown below; the action-native controller remains stronger in both subgroups throughout the L8–L32 sweep.

Layer	Held-out group	N	SafeEff$(-g_J)$	SafeEff(c^A)	$c^A - (-g_J)$
L16	All blocked-valid	60	0.0011	0.0267	0.0256
	KBD	35	0.0021	0.0267	0.0246
	non-KBD	25	-0.0002	0.0267	0.0269
L20	All blocked-valid	60	-0.0011	0.0505	0.0516
	KBD	35	-0.0009	0.0497	0.0507
	non-KBD	25	-0.0013	0.0517	0.0530
L24	All blocked-valid	60	0.0003	0.0633	0.0630
	KBD	35	0.0007	0.0241	0.0234
	non-KBD	25	-0.0003	0.1181	0.1183
L32	All blocked-valid	60	-0.0018	0.0158	0.0176
	KBD	35	-0.0021	0.0163	0.0184
	non-KBD	25	-0.0013	0.0151	0.0164

Thus the controller ordering observed in the complete blocked-valid population is also present separately in KBD and non-KBD cases for Qwen and Mistral. For Mistral, the same ordering is additionally preserved when the 60 held-out cases are divided into three equal-sized strata by baseline action margin.

K Split-Half Stability of Control Subspaces

The low overlap between judgment- and action-control subspaces could in principle arise if each estimated space were itself unstable under finite-sample variation. All headline stability analyses therefore use Split B of the causal set. For each model, tested layer, target tool, and rank $k \in \{1, 2, 4, 8, 16\}$, train-family scenarios are split into two halves while preserving tool and family structure as closely as possible. We independently construct $U^{J,(1)}_{\ell,k}, U^{J,(2)}_{\ell,k}$ and $U^{A,(1)}_{\ell,k}, U^{A,(2)}_{\ell,k}$, and

Model	Layer	Safe-oriented $c^A \to J$	95% CI	Safe-positive
Qwen3-32B	L32	-0.0167	[-0.0458, 0.0125]	5/30
Qwen3-32B	L48	0.0500	[0.0125, 0.0875]	13/30
Gemma3-27B	L32	-4.3667	[-4.7250, -4.0333]	0/60
Gemma3-27B	L48	-8.5625	[-8.8451, -8.2740]	0/60
Mistral-Small-3.2-24B	L8	0.0277	[0.0196, 0.0357]	27/35
Mistral-Small-3.2-24B	L12	0.0357	[0.0268, 0.0446]	27/35
Mistral-Small-3.2-24B	L16	0.0670	[0.0571, 0.0768]	34/35
Mistral-Small-3.2-24B	L20	-0.2134	[-0.3938, -0.0286]	15/35
Mistral-Small-3.2-24B	L24	0.2094	[0.1500, 0.2701]	27/35
Mistral-Small-3.2-24B	L28	0.6147	[0.5888, 0.6411]	35/35
Mistral-Small-3.2-24B	L32	0.3040	[0.2522, 0.3567]	35/35

Table 8: **Reverse transfer from action-native control to explicit judgment is architecture- and depth-dependent.** Positive values shift the explicit judgment toward `BLOCK`; negative values shift it toward `ALLOW`.

compute

$$S_{JJ} = S(U^{J,(1)}, U^{J,(2)}), \qquad S_{AA} = S(U^{A,(1)}, U^{A,(2)}), \qquad S_{JA} = S(U^{J,(1)}, U^{A,(2)}),$$

using the overlap score from Equation 10. Qwen and Gemma use 30 random split-half repetitions per configuration; the Split B Mistral analysis uses 50 repetitions. At rank 16, each half contains roughly 20 train gradients per target tool.

At rank 16, averaged over target tools and tested layers, the results are:

Model	S_{JJ}	S_{AA}	S_{JA}	Random expectation
Qwen3-32B	52.94%	54.56%	1.86%	0.3125%
Gemma3-27B	47.33%	47.42%	4.70%	0.2976%
Mistral-Small-3.2-24B	49.81%	42.14%	1.71%	0.3125%

For Mistral, the Split B rank-16 layerwise J–A overlap decreases from 3.16% at L8 to below 1% at L24–L32, while the corresponding within-computation J–J and A–A reproducibility remains substantially larger. Across all three models, the within-computation split-half overlap is therefore much higher than the cross-computation overlap. The low judgment–action overlap is not explained simply by finite-sample instability of the estimated subspaces. The spaces are not completely orthogonal: Gemma, in particular, retains more cross-computation sharing, consistent with the larger late-layer overlap reported in the main text.

L Action-Controller-to-Judgment Reverse Transfer

We additionally test the reverse intervention $c^A \to J$. This supplementary analysis evaluates the held-out KBD subsets associated with the corresponding controller analyses ($N = 30$ for Qwen, $N = 60$ for Gemma, and $N = 35$ for Mistral), rather than the complete 60-scenario Split A population used in Figure 3. The action controller c^A is constructed only from train-family blocked-valid scenarios, conditioned on target tool, frozen, and injected into the held-out KBD judgment interface. The judgment endpoint is the original margin

$$m_J = \log p(\texttt{ALLOW}) - \log p(\texttt{BLOCK}),$$

and we report the safe-oriented judgment effect $-\Delta m_J$, so positive values indicate movement toward `BLOCK`. The intervention modifies the final judgment-prompt token at a single residual position and is norm matched to the corresponding held-out case's allowed-valid minus blocked-valid judgment-state displacement. The analysis uses the same layers and controller hyperparameters as the corresponding forward-transfer experiments.

The reverse-transfer pattern is not a symmetric double dissociation. Qwen is closest to a weak-transfer regime: L32 is statistically compatible with zero, while L48 has a small positive mean

effect. Gemma shows a qualitatively different pattern: the action controller produces a large and completely consistent ALLOW-oriented judgment shift at both tested layers. Mistral changes sign with depth, with positive effects at L8–L16, a negative effect at L20, and positive effects again at L24–L32. Thus action-native controllers can be null, safety-oriented, or oppositely oriented on the judgment interface. The result therefore qualifies control separation as an interface-dependent causal relationship rather than causal independence between judgment and action computations.