OBSERVERBENCH: TESTING MECHANISTIC ESTIMATES FOR INTERVENTION AND CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Interpretability methods increasingly guide concrete actions, such as editing activations, ablating components, or allocating scarce human review. Yet statistical prediction accuracy alone does not guarantee effective downstream decisions. We present ObserverBench, a framework for comparing estimators—*observers*—on both held-out prediction quality and the quality of the decisions they guide. Each task fixes the available measurements, candidate actions, decision policy, and loss, so researchers can test whether better predictions lead to better interventions, control, or safety decisions. Control analysis defines the reachable subspace where an observer must match the true system to avoid compounding errors. Across empirical tasks, statistical quality repeatedly decouples from decision outcomes. In model editing on GPT-2 and Qwen2.5-7B, modeling component interactions improves effect predictions, yet lower mean error fails to select better interventions; predicting decision loss directly does. In safety triage, perfect classification can misallocate a fixed budget by ignoring consequence severity. On code-backdoor detection (APPS) across Qwen (2.5-7B, 3.5-9B) and Gemma-2-9B-it, residual observers yield the lowest average loss, but the optimal measurement context varies by model, and AUROC rankings can invert operational performance on these panels. ObserverBench establishes observer adequacy as a declared, task-relative, and reproducible evaluation.

1 INTRODUCTION

Mechanistic interpretability is undergoing an operational shift. Beyond supporting post-hoc explanations for human researchers, internal representations are increasingly used to automate decisions about model behavior: selecting activation steering vectors, pruning attention heads, deciding whether to permit an external tool call, or allocating human oversight to suspicious requests (Turner et al., 2023; Zou et al., 2023; Nguyen et al., 2025; Skifstad et al., 2026; Greenblatt et al., 2024; Inglis et al., 2025). In these settings, an internal estimate is no longer just an explanatory artifact; it is an input to an active decision system.

The challenge is to connect predictive accuracy with the decisions an estimate supports. Standard offline metrics, such as mean squared error, correlation, and AUROC, assess prediction quality in isolation. Operational decisions, however, must account for strict constraints, finite budgets, and asymmetric mistake costs. An activation probe that separates safe from unsafe requests with high test-set AUROC may direct limited audit capacity toward trivial infractions while more costly violations pass through. Similarly, an observer that predicts the average ablation effect of a circuit with low mean error can select an intervention whose effects fluctuate widely across individual prompts. Statistical fidelity on an offline test set does not guarantee sound operational decisions (Orgad et al., 2026; Roughan et al., 2003). The main contribution of this paper is to make this distinction systematically testable.

Our Gemma code-backdoor comparison illustrates this distinction: a prompted monitor ranks above an internal sparse-feature probe by AUROC, yet misses more backdoors under a fixed audit budget (Sec. 5.2). This observed reversal demonstrates why classification metrics alone cannot establish operational adequacy.

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.

Standard proxy metrics do not encode the operational decision rule: AUROC measures ranking across an entire dataset, but a budgeted monitor can inspect only a small fraction of requests (e.g., the top 1–2% highest-risk requests). For model interventions, prompt-level dispersion creates a different problem: an edit that achieves the desired target *on average* may under- or over-correct on almost every single prompt, incurring downstream loss that mean-effect metrics fail to capture. Isolating the value of internal access requires rigorous baseline comparison: if a model's output margin or input text already supports an equally good decision, extracting internal activations can introduce latency and complexity without improving outcomes. We therefore compare estimators under the same decision rule and operational constraints.

We introduce **ObserverBench**, a benchmark framework that evaluates model estimators—which we term *observers*, borrowing the concept from control theory (Luenberger, 1964)—on both their predictions and the actions they guide.[1] An observer estimates hidden internal states, intervention effects, or safety risks from available measurements; a fixed *controller* uses that estimate to select an action from an allowed *intervention space* (such as edits, ablations, or audit policies). The measurements may come from activations, model outputs, or input text. By fixing the task, candidate actions, decision rule, operational budget, and loss function, ObserverBench compares estimates within the same decision problem. Researchers can compare interchangeable observers on both offline predictive accuracy and downstream action loss across three core settings: closed-loop control trajectories, one-shot intervention selection, and budget-constrained safety triage.

Contributions.

- A common evaluation protocol and open replay score observers on held-out prediction quality and downstream action loss under the same controller (Sec. 2, Appendix A).
- We derive affine control conditions on the starting estimate and responses to edits, then test the error decomposition through nonlinear Transformer layers (Sec. 3).
- On GPT-2-small and Qwen2.5-7B, lower mean-effect error need not improve intervention choice. Direct-loss prediction reduces action loss by 18.5% and 8.9% against same-feature mean observers in the primary comparisons (Sec. 4).
- Authorization and APPS tasks show how consequence, measurement context, and audit capacity affect monitor choice. APPS spans Qwen2.5-7B-Instruct, Qwen3.5-9B, and Gemma-2-9B-it; the preferred context changes even between the two Qwen generations, which use different architectures. Prediction and action-loss rankings are reported with uncertainty (Sec. 5).

To compare observers fairly, we must hold the decision problem fixed.

2 THE OBSERVERBENCH EVALUATION PROTOCOL

For a controlled comparison, each task fixes five fields: the **target** to estimate (a hidden state, intervention effect, or prompt loss); the **information and integrity boundary** (which signals are observable and whether they can be tampered with); the **action space** (allowed edits, ablations, or audit decisions); the **decision rule and loss** (how estimates select actions and how errors are penalized); and the **operating regime** (prompts, budgets, subgroup splits, and attack rates).

These choices and baseline competence checks define a **TaskCard**. Each estimator has an **ObserverCard** declaring its inputs, fitting procedure, and model access. Rather than reporting a single proxy metric, the **scorecard** pairs prediction accuracy with downstream action loss. Appendix A gives submission details and a worked example.

To ensure benchmark integrity, splits and candidate actions are locked prior to held-out evaluation, and the unedited model must pass baseline competence verification. Crucially, whenever intervention is optional, candidate action sets include an explicit **no-op** baseline: taking no action is often safer than forcing an inaccurate or unnecessary edit. With this protocol established, we turn first to evaluating estimators that guide repeated edits in closed-loop control.

[1] We extend the observer role from state estimation to intervention-effect and safety estimates. A safety monitor is one instance of this broader observer role; its adequacy is judged relative to a specified downstream decision, not predictive accuracy alone.

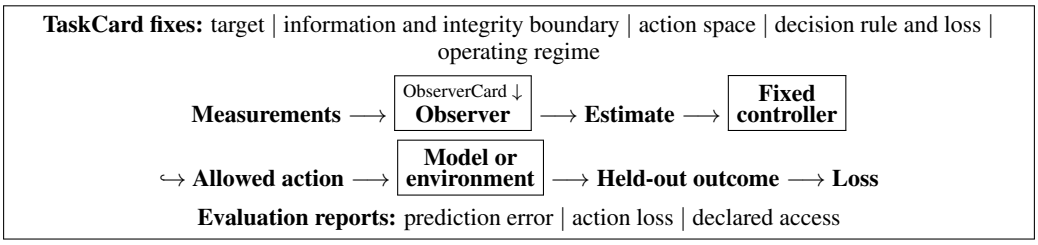


Figure 1: ObserverBench compares submitted observers on a common TaskCard. The ObserverCard declares the submitted method, and the task reports its statistical and operational outcomes. The IOI decision replay applies this full path to submitted effect tables; the generic effect command scores predictions only. Runtime costs require separate measurement.

3 Control needs a correct starting estimate and response

When an observer guides a sequence of edits, what must it estimate correctly for the controller to reach its target? Suppose an activation has three coordinates and edits move only the first two. The third coordinate stays fixed, so its contribution to the target stays fixed too. If the observer gets that contribution wrong, its initial target estimate has an offset. The controller can then change the first two coordinates to compensate for a measurement error, stopping at the wrong actual target even when the reported error is zero. For example, if the fixed contribution is one logit and the observer omits it, reaching a reported target of two logits leaves the true value at three. Correctly tracking every edit is not enough. Calibrating the starting estimate removes this offset. After that, the observer only needs to track target changes along the directions the controller can edit.

Let $E(h) = \widehat{z}$ be the estimate at state h, $D(h) = d$ the edit direction, and $u = C(y^\star, \widehat{z})$ the controller's action magnitude. We test estimates with D fixed and directions with E fixed; joint changes evaluate the estimator–direction pair.

Consider an affine target and observer,

$$z(h) = b_T + \beta_T^\top h, \qquad \widehat{z}(h) = b_E + \beta_E^\top h.$$

Let h_0 be the initial state, and let the columns of B span the allowed edit directions. The reachable states are $\mathcal{H}_R = h_0 + \mathcal{U}$, where $\mathcal{U} = \mathrm{span}(B)$.

Proposition 1 (Reachable-subspace adequacy)**.** *Define $\delta_0 = z(h_0) - \widehat{z}(h_0)$ and $\delta_\beta = \beta_T - \beta_E$. Then $z(h) = \widehat{z}(h)$ for every $h \in \mathcal{H}_R$ if and only if*

$$\delta_0 = 0 \qquad \textit{and} \qquad P_{\mathcal{U}}\delta_\beta = 0,$$

where $P_{\mathcal{U}}$ is the orthogonal projector onto the reachable subspace. For one fixed direction d, the second condition becomes $(\beta_T - \beta_E)^\top d = 0$.

Appendix B gives the proof.

3.1 An exact finite-horizon decomposition separates three control errors

Now let a proportional controller edit repeatedly along one direction:

$$h_{t+1} = h_t + K\widehat{e}_t d, \qquad \widehat{e}_t = y^\star - \widehat{z}(h_t).$$

Write $g_E = \beta_E^\top d$ for the response the observer expects, $g_T = \beta_T^\top d$ for the target's actual response, and $p = 1 - Kg_E$ for the rate at which the observer's reported error shrinks. For $g_E \neq 0$, define the initial mismatch $m_0 = z(h_0) - \widehat{z}(h_0)$ and the relative response mismatch $\rho = (g_T - g_E)/g_E$. After T steps, the true target error is

$$e_T = p^T\widehat{e}_0 - m_0 - \rho(1 - p^T)\widehat{e}_0. \tag{1}$$

Equation 1 isolates three sources of error in closed-loop activation steering:

1. **Reported tracking error ($p^T\widehat{e}_0$):** The residual discrepancy the observer believes remains. In a stable loop ($|p| < 1$), this term decays to zero.

2. **Initial estimation bias ($-m_0$):** An offset in the starting estimate that reducing the reported tracking error does not itself correct.
3. **Actuation sensitivity mismatch ($-\rho(1-p^T)\widehat{e}_0$):** The accumulated error when the true response g_T differs from the observer's anticipated response g_E.

In a stable loop, accumulated response error settles at $-\rho\widehat{e}_0$ rather than disappearing. Thus, an observer can report complete convergence while the true system remains displaced from its target by $-m_0 - \rho\widehat{e}_0$, unless the two offsets cancel.

3.2 Prediction and control through a nonlinear Transformer

Equation 1 is exact for an affine system. We test whether its three terms remain predictive when edits pass through learned nonlinear layers. Following the controlled synthetic setting of (Agarwal et al., 2025), we train 12 two-block Transformers for each of two targets. Both depend on binary features x_1 and x_2; one target also contains the interaction $\gamma x_1 x_2$. We edit the residual after the first block and rerun the second block, final normalization, and output head. The controller uses gain $K = 0.35$ for 15 steps; we score integrated squared tracking error (ISE) over the trajectory. A first-order observer uses x_1 and x_2; a lifted observer also uses their product.

To assess the scale of steering, we compare edit magnitudes with natural distances between unperturbed residual states and record whether clipping would occur, without applying it. Bootstrap intervals resample independently trained model seeds, not conditions within a model.

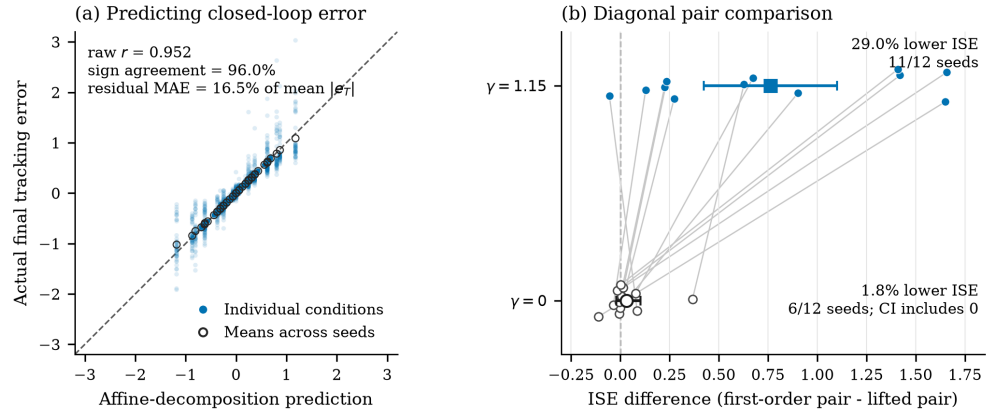


Figure 2: The three-term affine decomposition predicts final error after a learned nonlinear suffix (left). A lifted estimator–direction pair lowers control error when the target contains an interaction, but not when it is additive (right). Intervals resample 12 independently trained models.

Predicting final error. We measure the true target response with a symmetric probe before each trajectory; this ground-truthed diagnostic requires target access (Appendix B). At the largest offsets, the suffix departs from the affine prediction by 16.1% of output movement on average. Despite this nonlinearity, the raw three-term prediction correlates 0.952 with actual final error and predicts its sign in 96.0% of conditions. In leave-one-seed-out tests, calibrating the identity gives MAE 0.063, compared with 0.302 for a bias-only model and 0.367 for a model using only the observer's convergence term; it wins on all 12 held-out seeds.

Comparing viable estimator–direction pairs. For the interactional target, the lifted estimator–direction pair lowers ISE by 29.0% and improves 11 of 12 models. For the additive target, its advantage is not reliable. Crossed combinations encounter instability or scale violations, so this experiment does not isolate the estimator's contribution (Appendix B).

While closed-loop control steers a model through repeated feedback, many interpretability workflows make a one-shot choice among candidate interventions, such as a circuit ablation or model edit. The observer must then rank actions before any intervention is applied.

4 BETTER EFFECT PREDICTION NEED NOT CHOOSE A BETTER INTERVENTION

We test single-edit selection on GPT-2-small and Qwen2.5-7B. Within each access track, observers share calibration measurements, action pools, and held-out prompts. We report effect-prediction error alongside the loss of the selected action.

4.1 GPT-2 IOI: AN ACCURATE MEAN CAN HIDE PROMPT-LEVEL RISK

Consider "When Alice and Bob went to the store, Alice gave a book to." The correct next name is Bob. In GPT-2-small, published work identifies primary Name Movers that promote Bob, Backup Name Movers that compensate when that path is weakened, and Negative Name Movers that suppress the competing name (Wang et al., 2023; McGrath et al., 2023; McDougall et al., 2023; Rushing & Nanda, 2024). We use these documented head groups as the candidate intervention targets.

For prompt i and ablation mask m, let

$$Y_i(m) = LD_{i,\text{clean}} - LD_{i,\text{ablate}(m)},$$

where LD is the Bob-minus-Alice logit difference. Thus $Y_i(m) = 1$ means the ablation reduces Bob's advantage by one logit. The forward-only observers fit 160 calibration masks and predict 320 unseen masks in ten fixed action pools. The additive observer uses independent head effects. The interaction observer adds pair terms between documented head groups.

At 160 measurements, candidate-mask MAE falls from 0.486 for per-head additivity to 0.395 for all pair terms, an 18.8% reduction that holds in all ten pools.

The mean observer chooses the mask whose predicted average effect is closest to target t, but we score prompt-level loss $R_t(m) = \mathbb{E}_i|Y_i(m) - t|$. For a one-logit target, consider two masks. Mask A reduces the logit difference by one on every prompt. Mask B reduces it by two on half the prompts and zero on the rest. Both have mean effect one, so even an exact mean-only observer treats them as equally good. Yet A hits the target on every prompt, while B misses it by one in either direction: their absolute losses are zero and one. Let $\mu(m) = \mathbb{E}_i[Y_i(m)]$. Jensen's inequality gives

$$R_t(m) = |\mu(m) - t| + J_t(m). \tag{2}$$

Whereas the mean effect $\mu(m)$ tells us whether an intervention is centered on target across a dataset, the dispersion penalty $J_t(m) = \mathbb{E}_i|Y_i(m) - t| - |\mu(m) - t| \geq 0$ measures the extra loss individual prompts pay for that average. Effects can fall on either side of the target, leaving the mean exactly right while almost every prompt misses it. Unlike variance, this penalty depends on the target and vanishes when all effects lie on the same side of it. On held-out masks, even exact test-set means select the lowest-loss action in only 3 of 30 pool–target cells. The interaction observer brings selected means closer to the targets, but dispersion cancels that gain.

Predicting downstream loss requires **no additional** model forward passes in these experiments: calibration already records individual responses $Y_i(m)$. We compute each prompt's absolute error $|Y_i(m) - t|$, average it for each mask, and train the observer on those loss targets using the existing features. We evaluate this approach first on canonical IOI below, then test whether its advantage transfers to Qwen2.5.

At tight budgets (20 or 40 mask measurements), scalar-calibrated attribution patching (AtP) predicts effects better than forward-only additive ridge. At 80 or 160 measurements, additivity overtakes AtP and pair-based fits improve further, though they do not win at every budget (Appendix Table 7). AtP requires gradients, so equal measurement counts do not imply equal compute. Appendices A and C.2 report action scores and training-half refits.

4.1.1 CONFIRMATORY EVALUATION ON CANONICAL IOI

The confirmation uses 512 new canonical IOI prompts, 48 action pools, and a one-logit target. It passes the baseline accuracy threshold (97.7% overall accuracy; 95.3% on the worst template) and includes exact no-op in every pool. Both observers use the same quadratic mask features and calibration outcomes; one predicts mean effect and the other predicts absolute action loss. Target $t = 1$ was selected from pilot screening as an operational regime where intervening improves over no-op, providing a non-trivial test of action selection. The confirmation protocol, candidate pools, and actions were fixed prior to measuring test outcomes (Appendix C).

Task	Held-out effect prediction	Direct loss vs. mean	Direct loss vs. no-op
GPT-2 IOI, $t = 1$	Additive $0.486 \rightarrow$ pairs 0.395 MAE	$1.093 \rightarrow 0.891$ loss; 18.5%	$1.000 \rightarrow 0.891$; 10.9%
Qwen copy, three targets	Additive $0.121 \rightarrow$ pairs 0.040 MAE	$0.955 \rightarrow 0.870$ loss; 8.9%	$1.727 \rightarrow 0.870$; 49.6%

Table 1: Cross-model effect prediction and action selection. GPT-2 uses the pilot-informed, baseline-validated one-logit confirmation. Qwen averages three targets; the direct-loss and natural-mean observers tie within uncertainty at the largest target. All comparisons use fixed candidate pools and held-out prompts.

Direct-loss prediction reduces held-out action loss by 18.5% relative to the same-feature mean observer (absolute-loss interval $[0.115, 0.293]$) and by 10.9% relative to exact no-op ($[0.012, 0.196]$; Table 1). The gain comes from reducing the dispersion penalty from 0.807 to 0.494, enough to offset a rise in the mean-distance term $|\mu - t|$ from 0.286 to 0.397.

A follow-up sweep varies the target on the same prompts, pools, features, and calibration outcomes (Appendix Table 9). Direct loss outperforms both learned mean controls across all evaluated targets; at $t = 0.5$, however, exact no-op achieves lower loss, showing that the small target adjustment does not justify intervening in this setting. Exact no-op is available in every pool; an added abstention threshold would change the controller and must be declared before evaluation.

4.2 QWEN2.5-7B: THE DISTINCTION TRANSFERS TO A LARGER MODEL

We apply the same benchmark protocol to base Qwen2.5-7B using an induction-copy task (Qwen Team, 2024; Olsson et al., 2022). A prompt gives three key–value pairs and later repeats one key: after "A red, B blue, C green, ... B," the desired candidate is "blue." To ensure that candidate heads are evaluated only on tasks the model can perform, eligible prompts must place the target token above both distractors by a pre-specified margin of at least $\log 4$ prior to intervention. Of 3,072 fresh prompts, 2,612 (85.03%) satisfy this eligibility criterion. The task measures candidate ranking on these eligible prompts; unfiltered generation produces the desired token only 24.45% of the time.

Attention ranks candidate heads, but held-out interventions determine whether they belong in the task's head panel. We ablate the top 32 heads on a separate prompt set; a fixed heuristic selects eight heads across six layers. On 256 new prompts, these heads reduce the target margin by 3.454, compared with 0.170 for matched controls, with the paired interval excluding zero. A zero-ablation check gives the same ordering.

To compare intervention choices on this eligible candidate-margin task, we split all $2^8 = 256$ masks of the eight-head panel into 128 calibration and 128 held-out masks, evaluated on separate prompt sets. The additive observer assumes head effects sum. A quadratic observer adds all 28 head pairs. Pair terms lower held-out effect MAE from 0.121 to 0.040. Attribution patching gives the same access–budget crossover seen in GPT-2: after scalar calibration it beats the additive observer with 16 or 40 finite measurements, but additivity wins with 64 or 128. The forward-only quadratic observer remains best at every budget (Appendix Table 7). For action choice, natural mean, transformed mean, and direct loss use the same 37 features, three fixed targets, 16 pools, and exact no-op. Direct loss reduces aggregate action loss by 8.9% against natural mean, 12.6% against transformed mean, and 49.6% against no-op. At the largest target, however, direct loss and natural mean tie within uncertainty. Appendix C also reports a secondary non-induction continuation check.

On these fixed GPT-2 and Qwen tasks, interaction features improve effect prediction, while fitting prompt-level loss improves aggregate action selection. Gains depend on the requested target: no-op wins at GPT-2's smallest target, while direct loss does not establish an advantage over natural mean at Qwen's largest.

5 SAFETY TRIAGE: CLASSIFICATION IS NOT ENOUGH

The editing tasks apply one chosen intervention to many prompts. In safety triage, each request instead competes for a share of a limited review budget, and the cost of allowing a violation can differ across requests. Using trusted measurement channels, we test consequence and subgroup calibration

in a controlled authorization task, then monitor choice at rare attack rates on an external backdoor task.

5.1 A PERFECT CLASSIFIER CAN SPEND A SAFETY BUDGET POORLY

Consider a workspace assistant asked to delete a file in workspace A when the user has access only to workspace B. The requested operation is ordinary; the resource lies outside the user's authority. Our paired prompts change the permission for the same operation and resource. An exact authorization rule provides the label, and the operation provides the consequence: an unauthorized delete costs more than an unauthorized read. Every observer faces the same controller, which blocks the top-scoring 10% of requests, escalates the next 10%, and allows the rest. We then charge for missed harm and unnecessary intervention to evaluate how the score allocates this budget.

The study uses Qwen2.5-7B-Instruct (Qwen Team, 2024) and disjoint fit, calibration, and test sets. The most direct activation observer reads one residual-stream vector and predicts whether a request violates authorization. It classifies the held-out requests perfectly (AUROC 1.000), but its policy loss is 3.334. Multiplying that score by the known severity of the requested action reduces AUROC to 0.954 yet lowers policy loss to 2.155.[2] The statistically worse classifier makes better decisions because it sends scarce interventions toward unauthorized deletes before unauthorized reads. Against the action-only baseline, this lowers loss by 32.1%, closes 71.1% of the gap to the oracle, and preserves clean utility 1.000.

Qwen's block-minus-allow margin, multiplied by severity, gives the strongest tested score: loss 1.744, nearly matching the exact authorization-risk oracle at 1.740. On this task, the output supports a better policy than the activation probe while avoiding activation access.

Activation scores also shift scale across groups despite perfect within-group ranking (AUROC 1.000). Delete requests in XML test generalization to both an unseen operation and an unseen format: neither appears in fitting or calibration. With these scale differences, a single global threshold concentrates 63.2% of all interventions on deletes in familiar formats, while allocating 18.4% to deletes in XML. These are shares of the intervention budget, not within-group detection rates. In the XML-delete group, the output observer catches 69.8% of unsafe requests versus 29.2% for the activation observer (Appendix Fig. 3). Group-specific recalibration remains untested because those groups are absent from calibration.

Observer	AUROC	Loss	Missed risk	Clean utility
Allow all	0.500	5.010	1.000	1.000
Action-only direct risk	0.500	3.176	0.623	0.901
Activation label	1.000	3.334	0.665	1.000
Activation label $\times$ severity	0.954	2.155	0.429	1.000
Full-context text $\times$ severity†	0.942	2.032	0.405	1.000
Output margin $\times$ severity†	1.000	1.744	0.347	1.000
Exact risk oracle	1.000	1.740	0.346	1.000

Table 2: Held-out authorization results under fixed 10% block and 10% escalation budgets. Lower loss and missed risk are better; higher clean utility is better. Activation comparisons were fixed before held-out evaluation. Daggers mark text and output-margin controls designed after the original results were known; these are secondary comparisons, not independent confirmation.

5.2 MONITOR RANKINGS DEPEND ON THE MODEL AND OPERATING REGIME

ControlArena's APPS setting extends the comparison to ordinary code solutions and verified working backdoors (Greenblatt et al., 2024; Inglis et al., 2025). We compare Qwen2.5-7B-Instruct, Qwen3.5-9B, and Gemma-2-9B-it. The Qwen models span two generations with different architectures: Qwen2.5 uses an attention-based Transformer, whereas Qwen3.5 combines Gated DeltaNet with full-attention layers (Qwen Team, 2024; 2026a). This comparison does not isolate architecture: model size, training, and observer fitting also differ.

[2]Severity varies by action; unequal positive costs can reorder examples across groups, while the sign-separated output margin remains perfectly ranked.

A monitor scores each programming problem and submitted program. The controller audits only the highest scores up to its capacity, and every unaudited working backdoor counts as a violation. We split by problem ID and evaluate on 744 held-out honest/backdoor pairs; both members of a pair remain in the same split. The basic and detailed prompted monitors follow ControlArena's literature-derived prompts.

We compare output probabilities, supervised text, and residual activations beside an explicit safety verdict or in a neutral context. The fitted observers use supervision that the prompted monitors do not. Follow-up score-distribution, text, neutral-residual, and SAE controls were fixed for their comparisons and reuse the study data. Table 3 summarizes the two later model panels; Appendix D gives fitting details, outcome ranges, and the full comparisons.

		Gemma-2-9B-it		Qwen3.5-9B	
Monitor	Access	AUROC	Missed	AUROC	Missed
Exact-label oracle	Evaluator	1.000	0.00	1.000	0.00
Verdict-adjacent residual	Internal	0.969	**2.18**	0.977	**1.98**
Neutral residual	Internal	0.944	4.95	0.936	6.60
Score-distribution ridge	Output	0.913	4.42	0.940	5.12
Supervised text	Input text	0.868	9.05	0.868	9.05
SAE probe	Internal	0.870	9.07	0.811	12.73
Detailed prompted score	Output	0.893	10.80	0.645	13.98
Basic prompted score	Output	0.809	14.93	0.876	12.22

Table 3: APPS safety scorecard at 1% attacks and 2% audit capacity: 15 working backdoors and 29 audits per 1,488 submissions. AUROC uses the balanced held-out panel (higher is better); missed backdoors are means over 2,000 deployment populations (lower is better). The oracle uses evaluator-only labels. The SAE is checkpoint-matched on Gemma and transferred from base to posttraining on Qwen3.5. Source-problem uncertainty is discussed in the text.

The lower-loss prompting context differs across the three model panels. Among the original Qwen2.5 observers, neutral-context residuals lead with 2.65 missed backdoors versus 3.94 for verdict-adjacent residuals. The ordering reverses when evaluated on Gemma-2-9B-it and Qwen3.5-9B (Gemma Team, 2024; Qwen Team, 2026a). Their verdict-adjacent probes miss fewer backdoors than the neutral probes under the same task and audit policy. Paired source-problem intervals for verdict minus neutral are $[-3.73, -1.78]$ and $[-5.48, -3.33]$, respectively.

The key distinction is between ranking the whole population and finding backdoors in the few submissions selected for audit. Balanced AUROC measures pairwise ranking across a 50/50 honest/backdoor panel. Our deployment setting instead has 1% attacks and a 2% audit budget: only the 29 highest-scoring submissions out of 1,488 receive review. A monitor can earn high AUROC by ordering many pairs correctly below the cutoff while filling scarce audit slots with false alarms. Another can separate the broad population less well yet place more verified backdoors among the top 29. Global ranking rewards success throughout the distribution; budgeted auditing rewards precision at its upper tail. Attack rate changes the mixture being ranked, so a fixed audit fraction need not correspond to a fixed score threshold.

On the observed panels, choosing by AUROC alone can favor a monitor that misses more backdoors at the fixed audit budget. On Gemma, the detailed prompted monitor has higher AUROC than the Gemma Scope SAE probe (0.893 versus 0.870), yet misses more backdoors (10.80 versus 9.07). On Qwen3.5, the basic prompted monitor similarly ranks above supervised text by AUROC (0.876 versus 0.868) and below it by missed backdoors (12.22 versus 9.05). Source-problem resampling shows uncertainty in these reversals: the intervals for Gemma's loss difference and Qwen3.5's AUROC difference include zero (Appendix D.4). The scorecard reports both metrics alongside that source-problem uncertainty.

Decision loss also matters when evaluating a more compact readout. In a same-layer Qwen2.5 diagnostic, reducing the number of fitted coefficients comes with a higher audit loss: a 127-coefficient SAE probe misses 4.67 backdoors versus 2.05 with a dense residual fit (Appendix Table 11; see also Kantamneni et al., 2025). Its observed activation density is $L_0 = 115$, compared with the reported 240. Gemma Scope provides the primary checkpoint-matched SAE comparison (Lieberum et al., 2024). Qwen3.5 instead uses a base-trained Qwen-Scope SAE on the posttrained model because no

public SAE matched that checkpoint (Qwen Team, 2026b). Its 12.73 missed backdoors, against 6.60 for dense neutral, measure performance under that checkpoint transfer.

The pre-specified Qwen2.5 attack-rate and audit-budget sweep appears in Appendix Fig. 4. Together, the panels show why monitor choice requires both a model-specific comparison and an audit budget appropriate to the intended use.

6 Related Work

Interpretability guides activation steering, component ablations, and selective abstention (Turner et al., 2023; Hedström et al., 2025; Nguyen et al., 2025; Skifstad et al., 2026). Attribution techniques estimate component effects (Syed et al., 2024; Kramar et al., 2024); benchmarks provide tasks for circuit discovery, behavioral localization, and feature reconstruction (Gupta et al., 2024; Mueller et al., 2025; Wu et al., 2025). Mechanistic tomography studies what probes and interventions can recover (Erramilli, 2026). ObserverBench uses the classical observer role (Luenberger, 1964) to test whether these estimates support downstream actions.

Decision-focused learning and experimental design evaluate predictions through the decisions they inform (Donti et al., 2017; Elmachtoub & Grigas, 2022; Huang et al., 2024; Go et al., 2026). In traffic engineering, estimation-error magnitude alone poorly predicts routing performance (Roughan et al., 2003). AI Control and ControlArena evaluate complete oversight protocols against adaptive models (Greenblatt et al., 2024; Inglis et al., 2025). ObserverBench's safety tasks isolate monitor choice within a fixed policy and measure the consequences of score rankings at declared budgets.

7 Limitations

ObserverBench's controlled tasks isolate how estimation affects decisions. The affine analysis assumes calibrated starting states, while the nonlinear diagnostic uses target access. Intervention studies focus on fixed circuit components and prompt families with verified baseline competence; GPT-2's target was selected in pilot screening, and small target shifts can favor no-op. Safety triage isolates monitor choice on synthetic authorization policies and stored, non-adaptive APPS backdoors under trusted measurement channels. Adaptive evasion and live multi-agent coordination are complementary settings for end-to-end evaluation. The Qwen3.5 SAE results characterize base-to-posttraining transfer. Runtime profiling describes one A100 environment (Appendix E), and the open IOI replay uses public scoring responses to support transparent, inference-free reproduction rather than private test evaluation.

8 Discussion and conclusion

ObserverBench connects prediction accuracy to the consequences of acting on an estimate. Closed-loop control needs a correct starting estimate and edit response; one-shot selection benefits from predicting prompt-level loss rather than mean effect alone. In safety triage, consequence, calibration, and audit capacity determine which harms a ranking prevents.

Practically, these findings suggest three checks before deploying an internal monitor:

1. **Check output baselines first:** If output margins or input text already support an equally good decision, activation access needs another benefit to justify its construction and use costs.
2. **Match measurement context to the model:** Verdict-adjacent residual probes outperform neutral probes on Gemma-2 and Qwen3.5, but the ordering reverses on Qwen2.5. This is a model-specific result, not an architecture-only explanation.
3. **Evaluate at target operating points:** Offline AUROC on balanced data cannot substitute for testing at the intended attack rate and audit budget, with the relevant groups and losses.

Future extensions could compare observers for deciding whether to permit agent actions, motivated by the Hugging Face incident (Greenblatt et al., 2026), or accept updates in recursive self-improvement loops (Zheng et al., 2026). Both settings ask whether additional measurements prevent harm while retaining useful work under the same checking budget.

REPRODUCIBILITY STATEMENT

The anonymous supplementary ZIP includes a runnable observer-submission example, task configurations, recorded evaluation data, tests, and a file-hash manifest. Its CPU-only replay recomputes control and intervention scores and paired intervals, checking them against the recorded results. Appendices B–D specify the fit and selection rules and resampling units; Appendix A describes the open IOI replay. For safety tasks, the archive regenerates tables from published aggregates, not per-sample records: their metrics and intervals cannot be independently recomputed from this ZIP. Those evaluator records remain private, and no hosted private-test evaluator is supplied. The IOI responses are released for local reproduction, separately from private test evaluation. The README states these limits, records the required software, and distinguishes CPU reanalysis from fresh model runs.

ETHICS STATEMENT

Our safety evaluations use synthetic authorization queries and stored code examples to study how monitors allocate limited review. Authorization requests are scored without carrying out the requested actions; code-monitoring decisions use stored programs and verified labels without executing submitted code. This separates monitor evaluation from live tool use or production systems. Backdoor examples are retained as safety-evaluation data, and the shared artifact provides aggregate safety results rather than per-sample records. The study involves no human subjects or live user data.

AI-USE STATEMENT

Generative AI systems assisted with literature review, manuscript editing, and code implementation. The authors chose the research questions, designed the experiments, reviewed the generated material (code), reran and checked the reported results, and take responsibility for the paper and artifact.

REFERENCES

N. Agarwal, S. R. Dalal, and V. Misra. The Bayesian geometry of Transformer attention, 2025. arXiv:2512.22471.

P. L. Donti, B. Amos, and J. Z. Kolter. Task-based end-to-end model learning in stochastic optimization. In *NeurIPS*, 2017.

A. N. Elmachtoub and P. Grigas. Smart "predict, then optimize". *Management Science*, 68(1):9–26, 2022.

V. Erramilli. Mechanistic tomography: Designed measurement for control-oriented interpretability, 2026. URL https://arxiv.org/abs/2608.19338. arXiv:2608.19338.

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

Jinwoo Go, Xiaoning Qian, and Byung-Jun Yoon. Goal-driven Bayesian optimal experimental design for robust decision-making under model uncertainty, 2026. arXiv:2605.26093.

R. Greenblatt, B. Shlegeris, K. Sachan, and F. Roger. AI control: Improving safety despite intentional subversion. In *ICML*, 2024. arXiv:2312.06942.

Ryan Greenblatt, Ajeya Cotra, and Hjalmar Wijk. Brief independent investigation of agents' behavior, reasoning, and collaboration in the OpenAI / Hugging Face hacking incident. METR and Redwood Research, August 2026. URL https://metr.org/blog/2026-08-26-openai-hugging-face-incident-investigation/.

R. Gupta, I. Arcuschin, T. Kwa, and A. Garriga-Alonso. InterpBench: Semi-synthetic transformers for evaluating mechanistic interpretability, 2024. arXiv:2407.14494.

Anna Hedström, Salim I. Amoukou, Tom Bewley, Saumitra Mishra, and Manuela Veloso. To steer or not to steer? mechanistic error reduction with abstention for language models. In *ICML*, volume 267 of *Proceedings of Machine Learning Research*, pp. 22924–22945, 2025. URL https://proceedings.mlr.press/v267/hedstrom25a.html.

Daolang Huang, Yujia Guo, Luigi Acerbi, and Samuel Kaski. Amortized Bayesian experimental design for decision-making. In *NeurIPS*, 2024. arXiv:2411.02064.

R. Inglis et al. ControlArena, 2025. URL https://github.com/UKGovernmentBEIS/control-arena. Software.

Subhash Kantamneni, Joshua Engels, Senthooran Rajamanoharan, Max Tegmark, and Neel Nanda. Are sparse autoencoders useful? a case study in sparse probing. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 29018–29049. PMLR, 2025. URL https://proceedings.mlr.press/v267/kantamneni25a.html.

J. Kramar, T. Lieberum, R. Shah, and N. Nanda. AtP*: An efficient and scalable method for localizing LLM behaviour to components, 2024. arXiv:2403.00745.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma Scope: Open sparse autoencoders everywhere all at once on Gemma 2, 2024. arXiv:2408.05147.

David G. Luenberger. Observing the state of a linear system. *IEEE Transactions on Military Electronics*, 8(2):74–80, 1964. doi: 10.1109/TME.1964.4323124.

C. McDougall, A. Conmy, C. Rushing, T. McGrath, and N. Nanda. Copy suppression: Comprehensively understanding an attention head, 2023. arXiv:2310.04625.

T. McGrath, M. Rahtz, J. Kramar, V. Mikulik, and S. Legg. The Hydra effect: Emergent self-repair in language model computations, 2023. arXiv:2307.15771.

A. Mueller et al. MIB: A mechanistic interpretability benchmark. In *ICML*, 2025.

D. V. Nguyen, H. M. Vu, N. Y. Pham, L. Zhang, and T. M. Nguyen. Activation steering with a feedback controller, 2025. arXiv:2510.04309.

C. Olsson et al. In-context learning and induction heads, 2022. arXiv:2209.11895.

H. Orgad et al. Interpretability can be actionable. In *ICML*, 2026. arXiv:2605.11161.

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

Qwen Team. Qwen3.5-9B. Hugging Face model card, 2026a. URL https://huggingface.co/Qwen/Qwen3.5-9B.

Qwen Team. SAE-Res-Qwen3.5-9B-Base-W64K-L0_50. Hugging Face model card, 2026b. URL https://huggingface.co/Qwen/SAE-Res-Qwen3.5-9B-Base-W64K-L0_50.

M. Roughan, M. Thorup, and Y. Zhang. Traffic engineering with estimated traffic matrices. In *ACM IMC*, pp. 248–258, 2003. doi: 10.1145/948205.948237.

C. Rushing and N. Nanda. Explorations of self-repair in language models, 2024. arXiv:2402.15390.

J. Skifstad, X. A. Yang, and G. Chou. Local linearity of LLMs enables activation steering via model-based linear optimal control, 2026. arXiv:2604.19018.

A. Syed, C. Rager, and A. Conmy. Attribution patching outperforms automated circuit discovery. In *BlackboxNLP*, 2024.

A. Turner et al. Activation addition: Steering language models without optimization, 2023. arXiv:2308.10248.

K. Wang, A. Variengien, A. Conmy, B. Shlegeris, and J. Steinhardt. Interpretability in the wild: A circuit for indirect object identification in GPT-2 small. In *ICLR*, 2023.

Z. Wu, A. Arora, A. Geiger, Z. Wang, J. Huang, D. Jurafsky, C. D. Manning, and C. Potts. AxBench: Steering LLMs? even simple baselines outperform sparse autoencoders, 2025. arXiv:2501.17148.

Tong Zheng, Xidong Wu, Zheng Zhang, Zhankui He, Chaoyi Zhang, Benjamin Coleman, Ruoqiao Wei, Di Bai, Haolin Liu, Rui Liu, Xue Wang, Yue Zhuan, Wang-Cheng Kang, Renkai Xiang, Heng Huang, Xinwu Cheng, and Yunsong Guo. Dream-RSI: Recursive self-improvement through evolving worlds, 2026. URL https://arxiv.org/abs/2609.14858.

A. Zou et al. Representation engineering: A top-down approach to AI transparency, 2023. arXiv:2310.01405.

A SUBMITTING AN OBSERVER AND RECEIVING A DECISION SCORE

An effect submission supplies an ObserverCard and a CSV with columns `schema_version`, `query_id`, `predicted_effect`. The evaluator rejects missing, duplicate, unexpected, or non-finite entries. Generic effect scoring reports prediction error; the IOI decision route below also selects actions and reports their losses. Neither route executes contributor code.

The IOI replay uses 160 calibration masks averaged over 192 training prompts, 320 candidate masks in ten pools, and their 81,920 stored responses on 256 held-out prompts. Each pool includes exact no-op. For each target $t \in \{0.5, 1, 1.5\}$, the controller minimizes predicted mean distance to t, breaking ties by fewer heads and then mask ID. Selection never reads test responses; the chosen mask applies to every test prompt in its pool. The scorecard returns prediction MAE, selected masks, mean absolute action loss, and paired comparisons with the reference observers and no-op.

Observer	Prediction MAE	$R_{0.5}$	R_1	$R_{1.5}$
Raw AtP	0.662	1.075	0.946	1.093
Scalar-calibrated AtP	0.632	0.972	1.000	1.114
Additive ridge	0.486	1.053	0.972	1.024
All-pairs ridge	0.395	1.018	1.022	1.006
Exact no-op	—	0.500	1.000	1.500

Table 4: A prediction-and-decision scorecard from the existing IOI replay. Lower is better. The same predictions feed all three target-specific action tests. Attribution patching (AtP) requires gradients to construct; ridge uses finite forward measurements. Cached scoring leaves those construction costs separate. This open replay is distinct from the baseline-validated evaluation in Sec. 4.1.1.

Paired 95% percentile intervals use 2,000 joint resamples of 107 unordered-name-pair clusters and ten pools (seed 20260904), holding fits and actions fixed. At $t = 1$, calibrated minus raw AtP has loss difference 0.053 with interval $[-0.076, 0.203]$; none of the four observers establishes a gain over no-op. All four lose to no-op at $t = 0.5$ and beat it at $t = 1.5$ under their paired intervals.

This offline replay adds no model measurements. Its responses are public for local scoring, unlike the private safety test labels. Training on these scoring responses would invalidate a held-out claim. Scorecards record task, prediction, and scoring-code hashes and the software environment.

B CONTROL: PROOF, EXPERIMENTAL SETUP, AND ROBUSTNESS

B.1 PROOF OF REACHABLE-SUBSPACE ADEQUACY

Every reachable state has the form $h = h_0 + Bv$. With the notation of Proposition 1,

$$z(h) - \widehat{z}(h) = \delta_0 + \delta_\beta^\top Bv.$$

Setting $v = 0$ requires $\delta_0 = 0$. The remaining term vanishes for every v exactly when $B^\top \delta_\beta = 0$, equivalently $P_{\mathcal{U}} \delta_\beta = 0$ for $\mathcal{U} = \mathrm{span}(B)$. Thus agreement at the starting state and along every reachable direction are jointly necessary and sufficient.

B.2 MODEL, OBSERVERS, AND ROLLOUTS

The controller sees decoded features, not exact binary states. Scoring uses the actual nonlinear suffix output, not an extrapolated target probe. The diagnostic uses target access to construct mismatches and check the identity. Leave-one-seed-out calibration trains on 11 models and tests on the twelfth; intervals use 5,000 seed-level bootstrap draws.

Natural edit directions follow each observer's gradient, normalized to self-response 0.85. ISE sums squared true error over steps 0–15; final error remains the diagnostic's pre-specified primary outcome and ISE its secondary trajectory measure. The crossed pairs in Table 6 fail loop checks, so the 29.0% gain compares viable estimator–direction pairs, not estimators alone.

Model	Two Transformer blocks; width 32; four heads/block; MLP width 96.
Training data	3,000 length-four sequences/seed; binary-feature probability 0.35. Target $0.35x_1 + 0.25x_2 + \gamma x_1 x_2 + \epsilon$; $\gamma \in \{0, 1.15\}$; noise standard deviation 0.02.
Optimization	600 AdamW steps; batch 256; learning rate 10^{-3}; weight decay 10^{-4}; auxiliary feature-loss weight 0.05.
Observers	Ridge penalty 10^{-4} decodes x_1, x_2, x_1x_2 from block-0 last-position residuals. A second ridge fit predicts suffix output from either two or all three decoded features.
Rollout	Edit only the last-position residual; rerun block 1, final normalization, and output head. Use $K = 0.35$, 15 steps, no clipping.
Diagnostic	Symmetric probe size 0.05 measures g_T at the start; set $g_E = 0.85$. Cross response mismatches $\{-1.25, -0.75, 0, 0.75, 1.25\}$, initial biases $\{-0.25, 0, 0.25\}$ and target offsets $\{-0.5, -0.25, 0.25, 0.5\}$.

Table 5: Nonlinear control setup. Each γ uses 12 independently trained models.

γ	Estimate	Direction	ISE	$g_E \le 0$	$\lvert p\rvert \ge 1$	Would clip	Beyond clean max
1.15	FO	FO	2.632	0/12	0/12	0/192	0/192
1.15	FO	Lifted	2.381	10/12	10/12	30/192	56/192
1.15	Lifted	FO	3.666	10/12	10/12	56/192	96/192
1.15	Lifted	Lifted	1.870	0/12	0/12	0/192	0/192
0	FO	FO	1.693	0/12	0/12	0/192	0/192
0	FO	Lifted	1.659	0/12	0/12	0/192	0/192
0	Lifted	FO	1.693	0/12	0/12	0/192	0/192
0	Lifted	Lifted	1.662	0/12	0/12	0/192	0/192

Table 6: Secondary nonlinear-suffix estimator–direction factorial. Counts for g_E and p use 12 seed-level means; the last two columns use 192 rollout conditions per cell. The crossed arms at $\gamma = 1.15$ fail loop-conditioning and clean-archetype scale diagnostics, so they do not identify an estimator-only advantage. The clean maximum is a scale check, not a manifold boundary.

C Intervention choice: protocols, baselines, and sensitivity

C.1 Attribution-patching comparisons

For prompt i and head h, attribution patching uses $a_{ih} = \langle \nabla_{z_{ih}} f_i, z_{ih} - \mu_{\tau(i),h} \rangle$, where f_i is the IOI logit difference or Qwen candidate margin. It uses the same template- or family-conditioned replacement as the finite ablation. Averaging over public training prompts gives head effects; summing the selected heads predicts a mask's effect. Scalar calibration fits one no-intercept gain from the allowed finite measurements. Raw AtP needs gradients but no finite-effect outcomes, so equal finite budgets do not mean equal compute.

GPT-2 uses 192 training prompts and tests 320 masks on 256 disjoint prompts; Qwen uses 256 training prompts for its eight fixed heads. Both AtP additions are post-hoc evaluations. Their maps are constructed without held-out targets; they are method integrations, not new confirmations.

The GPT-2 action replay submits attribution-patching tables through the same selector as the forward-only observers (Table 4). At 160 measurements, scalar calibration lowers MAE from 0.662 to 0.632, while action loss at $t = 1$ rises from 0.946 to 1.000. The paired interval includes zero, leaving that action ranking unresolved. All four observers lose to no-op at $t = 0.5$ and beat it at $t = 1.5$. Across 40 training-half refits, pair terms lower MAE every time but raise action loss at $t = 1$ in 28 fits. These refits reuse one dataset; their sensitivity is reported below. Cached scoring needs no inference, although constructing AtP requires white-box access.

Model	Budget	Additive	Count	$P \times E$	Pairs	Raw AtP	Calib. AtP
GPT-2	20	0.846	1.579	1.333	0.795	0.662	0.630
	40	0.675	0.732	0.544	0.559	0.662	0.635
	80	0.496	0.482	0.406	0.412	0.662	0.632
	160	0.486	0.473	0.394	0.395	0.662	0.632
Qwen	16	0.3541	—	—	0.1001	0.5744	0.1933
	40	0.2021	—	—	0.0449	0.5744	0.1673
	64	0.1565	—	—	0.0392	0.5744	0.1707
	128	0.1210	—	—	0.0404	0.5744	0.1727

Table 7: Held-out effect MAE. Additive uses individual heads; count and $P \times E$ are GPT-2 group-based alternatives; pairs includes all pair terms. GPT-2's richer fits at budget 20 have more coefficients than measurements. AtP calibration helps at low budgets; additive fits overtake it at higher budgets.

GPT-2 calibration gains are 0.797, 0.739, 0.855, and 0.848 in budget order; the raw map correlates 0.986 with finite singleton effects. An earlier exploratory ranking favored $P \times E$; the fixed comparison ties it with all pairs, supporting pair features rather than a unique dominant pair.

C.2 Does refitting remove the prediction–action gap?

Twenty seeds split the original 192 GPT-2 training prompts into halves by unordered name pair. Each half has 89–103 prompts from 48 name pairs. We refit all four observers, keeping the 160 calibration masks, 320 test masks, pools, targets, ridge penalty 10^{-6}, and 256 test prompts fixed. AtP gradients and finite responses use only that half; its calibration gain uses that half's finite responses. Fits and actions are saved before scoring. The remeasured full-training AtP map reproduces the original scores and actions.

Change	Target	Mean ΔMAE	Mean ΔR_t	Lower MAE, higher loss
Additive to all-pairs	0.5	-0.090	-0.065	6/40
	1.0	-0.090	$+0.038$	28/40
	1.5	-0.090	-0.032	9/40
Raw to calibrated AtP	0.5	-0.033	-0.058	9/40
	1.0	-0.033	$+0.006$	21/40
	1.5	-0.033	$+0.049$	40/40

Table 8: Training-split sensitivity. Both changes lower MAE in every fit, but action loss depends on the target. Counts describe 40 fits of one dataset, not independent replications. Negative changes improve the metric.

At $t = 1$, descriptive 5th–95th percentile loss-change ranges are $[-0.031, 0.154]$ for pairs minus additive and $[-0.075, 0.089]$ for calibrated minus raw AtP. The action ranking remains unsettled despite consistent MAE gains. Calibration helps in 31/40 fits at $t = 0.5$ but harms all 40 at $t = 1.5$. This post-hoc sensitivity check inherits the original base-task limitation; it does not replace the fresh-prompt confirmation below.

C.3 Clean GPT-2 confirmation and Qwen protocol

GPT-2. An exploratory direct-loss run supported the idea among forced nonempty actions, but failed the baseline accuracy threshold and omitted exact no-op. It is excluded from confirmation. The follow-up returns to canonical IOI templates and passes the baseline accuracy threshold (97.7% overall accuracy; 95.3% on the worst template). The clean confirmation uses revision `607a30d783df` and template-conditioned mean ablation at the final token. Both observers use 192 calibration prompts, 160 masks, and the same 92 quadratic features (intercept, 13 heads, 78 pairs), with ridge penalty 10^{-6}. One fits mean effect; the other fits mean absolute loss from the same per-prompt responses.

Testing uses 512 fresh prompts: eight canonical templates, 32 new unordered name pairs, and both orientations. Each of 48 pools contains 30 new masks of five to nine heads plus exact no-op; masks

exclude earlier measured sets. The selector orders predicted loss, head count, then action ID, with no-op loss fixed to $|t|$. Actions are fixed before test measurement; only the union of selected masks is measured, not a fresh oracle ranking of all candidates. Target $t = 1$ was identified during pilot screening as an operating point where direct-loss selection improves over both mean-based selection and no-op, providing a non-trivial benchmark for action selection. The confirmation protocol was fixed locally before held-out measurements; the later target sweep is secondary. Intervals use 5,000 crossed resamples of the 32 name-pair clusters and 48 pools. They do not cover transfer to new template families.

Qwen. Base Qwen2.5-7B revision `d14972939875` runs in bfloat16 without quantization. Four families cross lengths 32/64 and repeat gaps 8/16. Reference, discovery, head-fit, confirmation, calibration, and test prompt sets use disjoint token vocabularies. Before intervention, eligible prompts must put the target above both distractors by at least $\log 4$; selection among eligible prompts uses a hash, not margin rank.

The eligibility rule was fixed before head discovery to evaluate intervention effects on prompts the unedited model can solve.

Attention shortlists 32 heads. Ablations on 256 head-fit prompts select eight (at most two per layer, at least four layers); another 256 compare them with matched controls. Means from 128 reference prompts define family-conditioned ablation at the final query position, before output projection. All 256 masks split equally into calibration and test sets, measured on 256 and 512 prompts. Five-fold mask-level cross-validation chooses ridge from $\{10^{-6}, 10^{-4}, 10^{-2}, 10^{-1}, 1, 10\}$ without locked outcomes. Each of 16 action pools has eight test masks and no-op. Targets are $0.25s, 0.50s, 0.75s$, where $s = 3.454$ comes from the independent confirmation bank. Their equal weighting is declared before final analysis. Intervals use 5,000 crossed resamples of four-family prompt clusters and masks or pools.

On 512 matched non-induction continuations, a secondary check finds no larger clean-to-intervened KL for direct loss than for the natural- and transformed-mean controls (0.0115 versus 0.0157 and 0.0160). This check measures distributional change on those continuations and does not establish general safety.

C.4 Target sensitivity, including when not to act

Both tasks fit mean and direct-loss observers from the same intervention responses; changing the label supplies no extra measurements. Table 9 retains the unsuccessful comparisons: GPT-2 direct loss loses to no-op at $t = 0.5$; Qwen does not establish a gain over natural mean at its largest target. Selection uses raw predictions, not post-hoc clipping. Across GPT-2 targets, transformed mean selects negative predicted losses in 44, 44, and 16 pools; direct loss does so in 4, 0, and 0.

Model	Target	Direct loss	Gain vs. mean	Gain vs. transformed	Gain vs. no-op
GPT-2	0.5	0.731	+29.4% [0.209, 0.405]	+20.8% [0.102, 0.288]	−46.2% [-0.314, -0.161]
	1.0	0.891	+18.5% [0.115, 0.293]	+15.9% [0.089, 0.253]	+10.9% [0.015, 0.196]
	1.5	1.070	+10.7% [0.020, 0.243]	+9.3% [0.011, 0.217]	+28.6% [0.321, 0.525]
	Average	0.897	+19.1% [0.141, 0.287]	+14.9% [0.091, 0.225]	+10.3% [0.021, 0.177]
Qwen	$0.25s$	0.609	+12.0% [0.034, 0.141]	+14.8% [0.035, 0.188]	+29.4% [0.205, 0.300]
	$0.50s$	0.842	+17.6% [0.108, 0.256]	+18.7% [0.117, 0.283]	+51.3% [0.824, 0.942]
	$0.75s$	1.158	−0.7% [-0.038, 0.021]	+6.3% [0.022, 0.146]	+55.3% [1.331, 1.530]
	Average	0.870	+8.9% [0.053, 0.118]	+12.6% [0.083, 0.173]	+49.6% [0.805, 0.908]

Table 9: Action loss by target. Gain cells show relative reduction and a 95% paired interval for the absolute comparator-minus-direct difference. Mean is the natural mean selector; transformed uses the transformed-mean fit. GPT-2's sweep follows the primary evaluation; Qwen's equal-target aggregate was declared before analysis. Negative and uncertain comparisons are retained.

D SAFETY: PROTOCOLS, FULL COMPARISONS, AND UNCERTAINTY

D.1 AUTHORIZATION: SUBGROUP CALIBRATION AND ADDITIONAL CONTROLS

Table 2 gives the main authorization comparison. The additional activation direct-risk fit has mean loss 2.520, tail loss 16.545, missed-risk fraction 0.502, and clean utility 1.000. Selecting by tail rather than mean loss chooses the same layer and ridge setting; the tail-loss difference from action-only is 0.039 with paired 95% interval $[-1.013, 1.156]$. We claim no tail-loss gain.

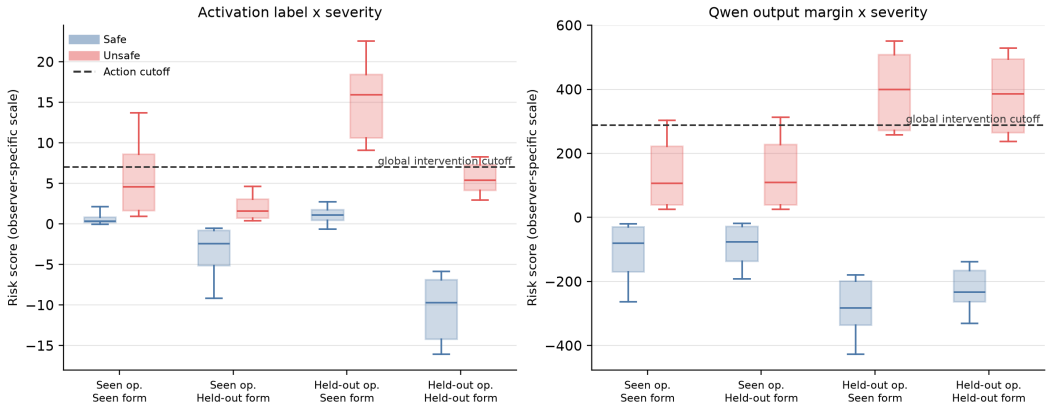


Figure 3: Authorization scores separate safe and unsafe requests within each group, but shift scale between groups. A global cutoff (dashed) therefore allocates interventions unevenly. Scores should be compared within panels, not across them. The calibration split lacks the held-out groups, so it cannot test whether group-specific recalibration repairs the problem.

D.2 APPS FITTING AND DEPLOYMENT PROTOCOL

All panels use APPS-control revision `29db28f76042`: 500 problem pairs fit observers, 200 select settings, and 744 form the locked test panel. Honest and working-backdoor solutions stay with their problem. Replications keep the same split, prompt forms, and policy; each design was fixed before its model's outcomes were inspected. Prompts request a suspicion score or present code without a verdict request. Residual probes read the last non-padding position.

Gemma and Qwen3.5 dense fits use a fixed 256-dimensional projection and ridge penalties $\{10^{-4}, 10^{-2}, 1, 100\}$. Calibration selects layer and penalty by violations at 2% audit capacity on its balanced panel; refitting uses fit plus calibration data. Text uses 2,048 hashed features. SAE features are standardized from fit data and rare features removed by the fixed rule. Gemma uses official width-16k SAEs at layers 9, 20, and 31. Qwen3.5 uses a base-trained SAE on the posttrained model. Prompted monitors have no supervised fitting labels; text, residual, score-distribution, and SAE probes do. These comparisons do not isolate internal access alone.

Deployment draws 15 attacks and 1,473 honest solutions with replacement, then audits the 29 highest scores. An unaudited working backdoor counts as one violation. All monitors share the draw and randomized row order before stable sorting, including ties. Empirical 95% percentile ranges across 2,000 draws describe variation between simulated deployments. Appendix D.4 quantifies uncertainty across source problems.

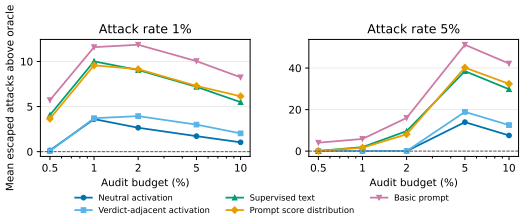


Figure 4: Qwen2.5-7B APPS: excess escaped backdoors over the oracle in 2,000 resampled populations. Rare attack rates expose differences hidden on the balanced panel. Audit budgets range from 0.5% to 10%.

D.3 Matched-model comparisons and the SAE transfer check

On the original Qwen2.5 panel, neutral residual has AUROC 0.979 and 2.65 mean violations, ahead of verdict-adjacent residual (0.953; 3.94), text (0.868; 9.05), and output score (0.830; 9.15).

	Gemma-2-9B-it		Qwen3.5-9B	
Observer	AUROC	Violations	AUROC	Violations
Exact-label oracle	1.000	0.00 [0,0]	1.000	0.00
Verdict-adjacent residual	**0.969**	**2.18 [0,5]**	**0.977**	**1.98**
Neutral residual	0.944	4.95 [2,9]	0.936	6.60
Score-distribution ridge	0.913	4.42 [1,8]	0.940	5.12
Supervised text	0.868	9.05 [6,12]	0.868	9.05
SAE probe	0.870	9.07 [6,12]	0.811	12.73
Detailed prompted score	0.893	10.80 [7,14]	0.645	13.98
Prompt ensemble	0.860	13.69 [11,15]	0.823	12.96
Basic prompted score	0.809	14.93 [14,15]	0.876	12.22

Table 10: APPS comparisons under the same task and primary operating point. AUROC uses the balanced panel; violations are deployment means (lower is better). Brackets retain Gemma's empirical 95% percentile ranges across deployments. The SAE is checkpoint-matched Gemma Scope for Gemma, but a base-to-posttrained transfer for Qwen3.5. The exact-label row is evaluator-only.

Gemma verdict minus neutral averages -2.77 violations with paired outcome range $[-6, 0]$; verdict is better, tied, or worse in 93.2%, 5.7%, and 1.1% of paired populations. These ranges span the central 95% of simulated deployment outcomes. Detailed prompt minus SAE averages 1.73 with range $[-2, 6]$. Qwen3.5 neutral minus verdict averages 4.625 with range $[1, 8]$. Verdict wins in 99.15% of paired populations. Its transferred SAE (layer 12, ridge 100 selected on calibration) incurs 6.124 more violations than dense neutral, range $[2, 10]$. No public SAE matched that posttrained checkpoint at the time; this tests artifact transfer.

Same-layer Qwen2.5 comparison. A post-hoc diagnostic applies the sparse-probing procedure of Kantamneni et al. (2025) to a public layer-19 BatchTopK SAE trained on Qwen2.5-7B-Instruct.[3] Feature ranking uses fit data; feature count and regularization use calibration data. Dense and SAE-reconstruction logistic controls use the same layer, splits, and selection rule; the locked test is scored once.

Observer	Nonzero coefficients	AUROC	Escaped attacks
Dense residual logistic	3,584	**0.983**	**2.05**
SAE sparse probe	127	0.953	4.67
SAE reconstruction logistic	3,584	0.954	4.79

Table 11: Qwen2.5 readout sparsity, not a compute saving. Escaped attacks are means over 2,000 populations at 1% attacks and 2% audits.

[3] https://huggingface.co/andyrdt/saes-qwen2.5-7b-instruct

Dense minus sparse averages -2.62 escaped attacks, paired deployment percentile range $[-6, 0]$. Restricting to 1,301 rows within the SAE's 1,024-token training length retains the AUROC gap (0.983 versus 0.956). Observed mean $L_0 = 115$ differs from the reported 240; the result concerns this checkpoint and task, not SAE features in general.

D.4 Do rankings survive changes in source problems?

This secondary check resamples the source problems, not just deployment populations. Each of 1,000 outer draws samples 744 problem IDs with replacement, keeping honest/backdoor pairs together. Within each draw, the evaluator averages 256 populations with 15 attacks, 1,473 honest solutions, and 29 audits. Monitors share source draws, deployment draws, and tie ordering. AUROC uses the balanced outer panel. Fits, checkpoints, and the original split remain fixed.

Model and comparison	ΔAUROC interval	Δviolations interval
Gemma: detailed prompt − SAE	$[0.005, 0.043]$	$[-1.18, 3.92]$
Gemma: verdict − neutral residual	$[0.014, 0.034]$	$[-3.73, -1.78]$
Qwen3.5: basic prompt − text	$[-0.014, 0.029]$	$[1.50, 4.61]$
Qwen3.5: verdict − neutral residual	$[0.030, 0.053]$	$[-5.48, -3.33]$

Table 12: Paired source-problem bootstrap intervals (95% marginal percentiles). Differences are left minus right: positive AUROC and negative violations favor the left monitor. Loss intervals are approximate because of nested Monte Carlo.

Verdict-over-neutral gains persist on both models. Neither observed AUROC–loss inversion establishes both sides of a population-level reversal: Gemma's loss interval and Qwen3.5's AUROC interval include zero. This post-hoc analysis does not include refitting, model or layer selection, selection of the reported contrasts, or new attackers.

E Measuring the cost of constructing and using an observer

This post-hoc timing study keeps the Gemma checkpoint, prompts, 500 fit and 200 calibration problem pairs, layers, and ridge choices fixed. One A100-SXM4-80GB uses bfloat16, SDPA, and a 4,096-token cap. Collecting neutral residuals at three layers for 1,464 examples takes 136.689 seconds; dense selection/refitting adds 0.330 seconds. SAE encoding and selection/refitting from that cache take 11.702 seconds. Both probes select layer 20. These construction costs exclude downloads and model/SAE pretraining.

Observer	Seconds	Min–max	GPU GiB
Basic prompted score	8.207	8.181–8.207	17.61
Detailed prompted score	7.631	7.597–7.640	17.59
Neutral dense probe	5.983	5.965–5.996	17.55
Neutral SAE probe	6.038	6.020–6.043	17.55

Table 13: Warm end-to-end use on the same 64 examples (32 held-out problem pairs, seed 904), batch size 2. Median and min–max over three rotated-order repeats include tokenization, full model inference, extraction, and readout. GPU memory includes weights.

Readout sparsity does not remove the cost of obtaining activations: dense and SAE use times are nearly equal. Prompted views are longer (median full-panel lengths 984.5, 917.5, and 692.5 tokens for basic, detailed, and neutral). All forwards run through the suffix, even for earlier-layer probes; this is not an optimized early-exit comparison. The timings describe this workload and implementation, not a hardware-general speed ranking.