LEARNING THE EVALUATOR: HOW LLMS ADAPT TO HIDDEN EVALUATION RULES

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models operate in review or compliance workflows where outputs are assessed over multiple interactions while the exact decision rule remains unknown. This can make direct adjustment harder at first, yet successive outcomes may reveal which outputs are more likely to succeed. This raises the question: can an LLM use repeated results to adapt specifically to the evaluator rather than simply improve with experience? We study this in a controlled sequential setting where the active rule remains hidden from the model but known to the experimenter, enabling exact scoring against the rule-specific optimum and comparison with a matched control. Across four model families, evaluator-dependent outcomes reduce regret relative to the active evaluator's optimum more than matched controls. On GPT and Gemini, behavior re-adapts after unannounced rule changes and information-sensitive actions emerge when they can improve later decisions; on Gemini, evaluator-specific adaptation also transfers beyond position-dependent evaluation. This paper contributes a study of evaluator-specific learning and shows that evaluation rules can become behaviorally learnable over time. The results show that hiding the active evaluator identity alone does not prevent repeated outcomes from becoming informative to an adaptive model, motivating evaluation designs that consider both detection quality and what interaction reveals over time.

1 INTRODUCTION

Evaluation has long been known to shape the behavior of those being evaluated, including in educational testing and strategic decision settings Alderson & Wall (1993); Wall & Alderson (1993); Ghalme et al. (2021). LLMs now increasingly operate in interactive agent and automated-evaluation settings, where outputs may be scored, verified, blocked, or monitored Liu et al. (2024); Zhou et al. (2024); Tan et al. (2025); Kale et al. (2026). Keeping the evaluator's exact rule hidden can make direct optimization against it harder Ghalme et al. (2021); Shao et al. (2025). Repeated interaction, however, creates an additional channel: outcomes can reveal properties of the evaluator and influence later actions Pan et al. (2024); Jiralerspong et al. (2026); Yao et al. (2025). As a result, behavior observed under one-shot evaluation need not describe how the same model behaves after repeated exposure Pan et al. (2024); Kale et al. (2026).

Existing work shows that LLM behavior can change under repeated feedback Pan et al. (2024) and monitoring Kale et al. (2026); Jiralerspong et al. (2026), while complementary work studies strategic response to partially observed decision rules, including black-box LLM guardrails and strategic classification Yao et al. (2025); Ghalme et al. (2021); Shao et al. (2025). What has not, to our knowledge, been isolated is whether evaluator-dependent outcomes causally move the same task-performing model toward the policy optimal for a particular hidden evaluator, beyond generic improvement or properties of the realized feedback sequence. We call an evaluator *behaviorally learnable* when evaluator-dependent outcomes cause later actions to move toward the policy that is optimal for that evaluator, beyond changes induced by repeated interaction alone. Here, learning refers to adaptation to the evaluation process and its decision criterion, not acquisition of task structure or factual knowledge. The claim is behavioral rather than cognitive: the experiments concern rule-specific changes in subsequent decisions, not what the model internally "knows" or represents.

Our experiments progressively separate evaluator-specific adaptation from generic improvement, the realized feedback sequence, fixed strategies, and the simple position-dependent structure of the

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.

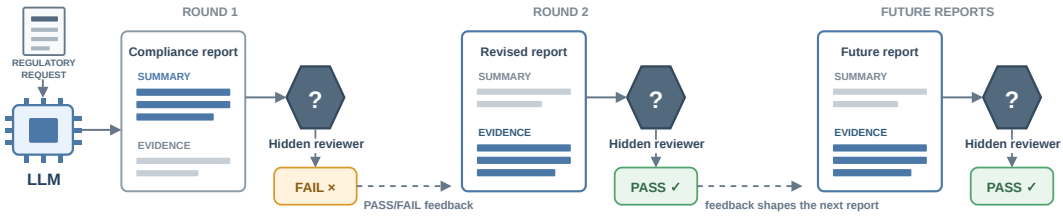

Figure 1: Repeated evaluation with a hidden reviewer. Binary outcomes can shape later outputs without revealing the review rule.

primary environment. Figure 1 illustrates the basic interaction: a model produces an output, receives only the review outcome, and carries that history into later decisions. In our experiments, feedback tied to the model's own actions produces larger reductions in regret than matched control feedback across GPT-5.4-mini, Gemini, DeepSeek, and Qwen. On GPT and Gemini, counterfactual scoring shows that the change is primarily rule-specific; on GPT, exact outcome replay nearly eliminates adaptation when the recipient's own action–outcome relation is broken. In additional experiments on GPT and Gemini, behavior re-adapts after an unannounced rule change and costly informative actions become more frequent when their information can improve later decisions; on Gemini, the effect also transfers to a structurally different task. These results are behavioral and controlled; they do not establish internal evaluator representations or degradation of deployed oversight.

Our contributions are threefold:

- **Evaluator-specific adaptation.** Repeated rule-dependent feedback moves later LLM decisions toward the active hidden policy beyond matched repeated feedback. Counterfactual scoring and matched outcome replay distinguish this effect from generic improvement and the realized outcome sequence. The core effect replicates across four model families.
- **Dynamic and information-sensitive behavior.** In additional experiments on GPT and Gemini, behavior partially re-adapts after unannounced rule changes, and costly informative actions become more frequent when the information remains useful for later decisions.
- **Structural transfer beyond position-dependent evaluation.** In a separate Gemini experiment, evaluator-specific adaptation transfers to fresh instances under a feature-dependent rule family with a different action representation.

2 Related Work and Positioning

Our question lies at the intersection of feedback-driven adaptation, adaptive monitoring and oversight, and strategic response to hidden decision rules. These literatures establish that evaluation can shape behavior, but do not identify adaptation to the optimum of a particular hidden evaluator.

Feedback-driven adaptation and reward optimization. Preference learning and RLHF use feedback to shape trained policies Christiano et al. (2017); Ouyang et al. (2022), while critique- and history-based methods can change later outputs without parameter updates Madaan et al. (2023); Shinn et al. (2023); Gou et al. (2024). Reward-model overoptimization, proxy exploitation, and reward hacking show how imperfect feedback can itself become an optimization target Rafailov et al. (2024); Miao et al. (2024); Laidlaw et al. (2025); Liu et al. (2025); Wang et al. (2026); Ferreira et al. (2026); reward-signal reliability is studied directly in Lambert et al. (2025); Yu et al. (2025). Most directly for our sequential setting, Pan et al. (2024) show that feedback loops can induce in-context policy refinement, including reward-hacking behavior that static evaluation does not expose. Targeted and inference-time strategies can likewise exploit properties of human or model feedback Williams et al. (2025); Khalaf et al. (2025). These results establish feedback-driven adaptation, but do not isolate adaptation to a particular hidden evaluator.

Evaluators, monitoring, and adaptive oversight. LLM judges, reward models, and agent evaluators can disagree, exhibit evaluator-specific biases, or expose exploitable failure modes Zheng et al. (2023); Kim et al. (2024a;b); Wang et al. (2024); Chen et al. (2024); Zhuge et al. (2025); Tan

et al. (2025); Raina et al. (2024); Tong et al. (2025); Huang et al. (2026). AI-control work studies oversight under intentional subversion and adversarial monitor awareness Greenblatt et al. (2024); Kale et al. (2026). Related scalable-oversight work asks whether weaker or fallible evaluators can reliably supervise stronger systems Burns et al. (2024); Recchia et al. (2026). Monitor performance can degrade when the acting model is aware of monitoring Kale et al. (2026), while hidden behaviors may require targeted auditing Sheshadri et al. (2026). Most closely related, Jiralerspong et al. (2026) show that frontier agents can infer the presence of hidden chain-of-thought monitoring from coarse blocking feedback. Monitor awareness alone, however, does not show that feedback identifies enough of a particular evaluator rule to improve later task decisions.

Hidden decision rules and strategic response. Strategic responses to evaluation predate LLMs, including in educational testing and inspection settings Alderson & Wall (1993); Wall & Alderson (1993); Avenhaus et al. (2002). Strategic-classification work more directly studies hidden or selectively revealed decision rules Ghalme et al. (2021); Shao et al. (2025). Modern LLM evaluation also studies actively selecting which evaluation outcomes to acquire Li et al. (2025b), dynamic rules for training safety reward models Li et al. (2025a), and sequential preference-feedback settings Tu et al. (2025). Evaluation-time policy switching in offline reinforcement learning concerns switching between learned policies rather than a changing evaluator rule Neggatu et al. (2025). Black-box guardrail reverse engineering is closer to our information structure, but trains a separate surrogate from repeated query–decision observations Yao et al. (2025). We instead study whether the *same task-performing LLM* turns a short evaluator-dependent history into better future actions against the latent rule. Unlike receiver-side bounded verification or one-shot review-aware composition, our object is adaptation through repeated evaluator-dependent feedback.

3 Research Questions and Experimental Framework

Evaluator learnability requires more than improvement: later behavior must become better suited to the hidden rule, rather than improving in a way that would help under any evaluator.

3.1 Research Questions and Experimental Logic

Our central question is whether repeated evaluator-dependent feedback can make a hidden evaluator behaviorally learnable to the same task-performing LLM. We address it through three linked questions. **RQ1 (evaluator-specific adaptation)** asks whether such feedback moves later decisions specifically toward the policy preferred by the active hidden evaluator, rather than producing only generic improvement or reflecting the realized sequence of feedback. **RQ2 (dynamics and information sensitivity)** asks whether this adaptation follows changes in the evaluator and whether behavior responds to whether information about the evaluator can still improve later decisions. **RQ3 (structural transfer)** asks whether evaluator-specific adaptation transfers when the task and evaluator depend on item features rather than on the position-dependent rules used for primary identification.

The experiments address these questions in an increasingly demanding sequence. We first use two opposing position-dependent evaluator policies for which the evaluator-specific optimum and regret can be computed exactly. Matched feedback controls, scoring each output under both candidate policies, and exact replay of outcome sequences test whether any improvement is specific to the active evaluator and to the model's own action–outcome relation. We then change the evaluator without announcement and manipulate whether costly information about it remains useful for later decisions. Finally, we replace the position-dependent evaluator family with feature-dependent rules and test fresh-instance transfer. Sections 4–6 report these three stages in turn.

3.2 Interaction and Threat Model

In each round, the model sees a new task instance together with the visible record of its earlier submissions and outcomes, chooses a task output A_t, and receives only PASS or FAIL. Let Y_t denote that binary outcome and $H_t = (A_1, Y_1, \ldots, A_{t-1}, Y_{t-1})$ the visible history. We write

$$A_t \sim \pi_G(\cdot \mid H_t), \qquad Y_t \sim P(\cdot \mid A_t, \Theta),$$

where Θ is the active evaluator policy.

We deliberately disclose the candidate evaluator family while hiding the active member. This isolates evaluator identification from open-ended hypothesis-class discovery and makes evaluator-specific optima, counterfactual utilities, and regret exactly identifiable. The experimenter knows Θ and the full mechanism; the generator sees the task, a qualitative description of the candidate family, the prior, and its history, but not the active identity or exact numerical likelihoods. This setup is related to strategic-response work on behavior under partially observed decision rules Ghalme et al. (2021); Shao et al. (2025). Prompts do not instruct the model to infer, probe, or exploit the evaluator. Our claims concern observable decisions, not internal representations; information-access and prompt audits are in Appendices J and L.

3.3 Controlled Evaluation Environment

We use the primary environment as an identification instrument: its finite structure makes evaluator-specific optima, counterfactual utilities, and regret exactly computable. We instantiate it as a controlled claim-selection task built from FEVEROUS and SciFact Aly et al. (2021); Wadden et al. (2020). A later feature-dependent transfer study then tests whether the effect extends beyond this positional geometry. Each benchmark-derived source pool supplies four supported and four unsupported claims carrying task values. The generator selects and orders four claims subject to $V(A) \geq R$, where R lies midway between the best supported-only and highest-value unconstrained selections. Unsupported content can therefore be necessary for high task value, creating a controlled utility–review-risk trade-off.

The evaluator inspects two claims without replacement and returns FAIL if either is unsupported. Audit feedback is generated for every well-formed output, while scored utility is zero below the value threshold; Appendix C reports the small number of value-infeasible submissions and the corresponding sensitivity analysis. The primary environment has two hidden evaluator policies that differ only in which positions they inspect more heavily. We call the earlier-position-biased policy FRONT and the later-position-biased policy BACK:

$$w_i^{\text{FRONT}} \propto e^{-\gamma(i-1)}, \qquad w_i^{\text{BACK}} \propto e^{-\gamma(n-i)},$$

with sequential sampling from the remaining claims. Their optima conflict, so an ordering favored by one policy can be poor under the other. Feasibility checks, policy-separation checks, and data-source details are in Appendix A.

3.4 Identifying Evaluator-Specific Adaptation

A decline in regret alone does not identify evaluator learning. In the primary eight-round setting, the model knows that either FRONT or BACK is active with prior $1/2$, but not which one. The active policy is fixed within an episode, and earlier submissions and outcomes remain visible.

We compare two task- and schedule-matched feedback processes:

$$Y_t \sim \begin{cases} P(\cdot \mid A_t, \Theta), & \text{REAL}, \\ \text{Bernoulli}(q_t), & \text{NULL}. \end{cases}$$

For each task instance, q_t is the mean exact pass probability over all feasible ordered outputs and is identical under both policies. In NULL, it depends on the task but not the submitted output or active evaluator. NULL therefore preserves task, schedule, outcome type, and repeated binary experience while removing evaluator-specific action–outcome dependence. It does not match the realized REAL outcome sequence. A separate exact-yoked condition, denoted YOKED, replays the exact PASS/FAIL sequence from a matched REAL trajectory while the recipient model generates its own actions; this tests whether the realized outcome sequence alone can explain the change. No-history and YOKED results are reported in Section 4 and Appendix C.

3.5 Exact Regret and Adaptation Measure

We quantify adaptation using exact evaluator-specific regret. Because PASS/FAIL is stochastic, we score submitted artifacts against the exact active-evaluator optimum:

$$U_\theta(A) = \mathbf{1}[V(A) \geq R]\, P_{\text{pass}}(A \mid \theta), \qquad A_\theta^* \in \arg\max_A U_\theta(A), \qquad R_\theta(A) = U_\theta(A_\theta^*) - U_\theta(A).$$

Table 1: **Threat model and information access.** The target LLM never receives the active evaluator identity, exact numerical likelihoods, oracle regret, or future outcomes. YOKED exactly replays a donor outcome sequence but is not evaluator-information-free.

Condition	Own history	Active θ	Exact likelihoods	Depends on own action?	Scientific role
No history	no	hidden	no	—	history ablation
NULL	yes	hidden	no	no	evaluator-uninformative repeated-feedback control
YOKED	yes	hidden	no	no	exact donor outcome-sequence robustness; donor stream can retain policy information
REAL LLM	yes	hidden	no	yes	target adaptive policy
Bayesian reference	yes	hidden	yes	yes	informed computational reference
Transparent control	yes	revealed	no	yes	known-policy capability ceiling
Oracle	—	revealed	full	—	exact task optimum

Thus $R_\theta(A) = 0$ is evaluator-optimal; for feasible actions, regret equals the success-probability loss to the optimum. For episode e, adaptation is the reduction in regret from the first two to the final two rounds,

$$D_e = \frac{R_{e,1} + R_{e,2}}{2} - \frac{R_{e,7} + R_{e,8}}{2}, \qquad \Delta_D = \mathbb{E}\big[D_e^{\mathrm{REAL}} - D_e^{\mathrm{NULL}}\big].$$

Our primary estimand is the REAL–NULL difference in early-to-late active-evaluator regret reduction; the primary hypothesis is $\Delta_D > 0$. Here, learning means adaptation to the evaluation process and its decision criterion, not acquisition of task ontology or factual knowledge. Inference preserves REAL–NULL pairing and source-pool dependence (Appendix K). Table 1 summarizes the information structure across conditions.

The primary hidden-evaluator study uses `gpt-5.4-mini-2026-03-17` with medium reasoning effort. Replications and later interventions use provider-native settings rather than post-hoc harmonization; exact snapshots, reasoning mappings, temperatures, retry rules, and structured-output schemas are listed in Appendix J.

4 Hidden Evaluators Become Behaviorally Learnable

The core analysis first checks that the model can act differently when the evaluator policy is revealed, then asks whether the same policy-specific adaptation emerges when evaluator identity is hidden and only feedback is observed.

4.1 Transparent Policies Induce Near-Optimal Specialization

We first verify that the model can specialize to the candidate policies when the active evaluator is revealed. Because FRONT and BACK have opposing optima, success requires policy-specific behavior rather than generic caution. We summarize specialization by policy-adaptation score (PAS), the mean within-policy minus cross-policy utility gap; its full definition is in Appendix D. This capability gate was run before the hidden-evaluator study and is not a matched baseline for its 59-pool panel. Across 40 source pools and 320 confirmatory decisions, $\mathrm{PAS} = 0.5482$ (95% clustered CI $[0.5201, 0.5793]$); the correct reversal occurs in all $40/40$ pools, and $158/160$ policy-conditioned decisions attain zero regret. Complete matrices are in Appendix D.

4.2 Rule-Specific Adaptation under Evaluator-Informative Feedback

We next hide evaluator identity and test whether evaluator-dependent feedback improves future decisions relative to matched repeated feedback without evaluator-specific action dependence. The primary GPT-5.4-mini experiment contains 472 matched REAL–NULL pairs of eight-round episodes: 7,552 decisions across 59 FEVEROUS and SciFact source-pool clusters. Schedules and visible interfaces are matched; only REAL preserves evaluator-dependent action–outcome information. The paired contrast is

$$\widehat{\Delta}_D = 0.2068, \qquad 95\%\ \mathrm{CI} = [0.1812, 0.2323].$$

The two-way clustered standard error is 0.01276 with 58 source-pool degrees of freedom. Thus informative feedback produces an approximately 20.7-percentage-point larger reduction in the exact feasibility-gated true-policy regret. Figure 2a shows the round-by-round trajectories.

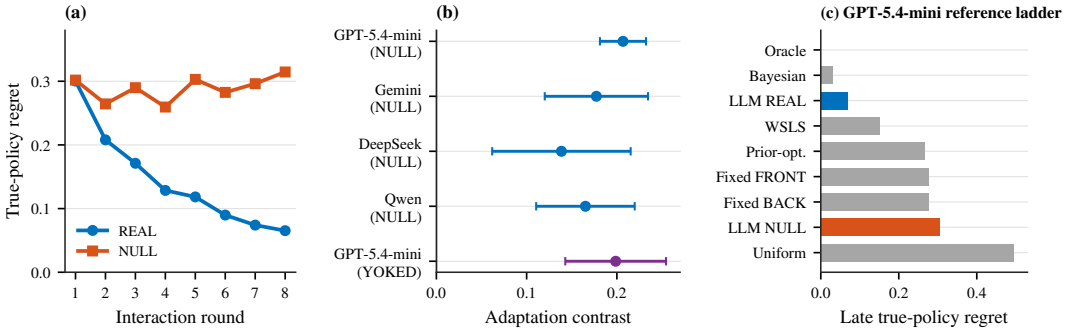

Figure 2: **Hidden-evaluator adaptation and reference policies.** **(a)** GPT-5.4-mini true-policy regret across interaction rounds under REAL and NULL. **(b)** REAL–NULL adaptation contrasts for GPT-5.4-mini, Gemini, DeepSeek, and Qwen, plus the visually distinct GPT exact-yoked REAL–YOKED contrast; bars show 95% confidence intervals. **(c)** Absolute late true-policy regret for GPT-5.4-mini and same-environment reference policies.

The same 64-pair hidden-evaluator design gives positive effects for Gemini 3.8 Flash, $\widehat{\Delta}_D = 0.1773$ (95% CI $[0.1201, 0.2345]$), DeepSeek V4 Pro, 0.1386 ($[0.0619, 0.2153]$), and Qwen 3.8-27B, 0.1652 ($[0.1106, 0.2198]$). Together with GPT-5.4-mini, these are robustness checks across four model families rather than pooled evidence. Figure 2b reports the four REAL–NULL estimates together with the exact-yoked contrast below.

4.3 Counterfactual Specificity, Outcome Yoking, and Reference Policies

A positive REAL–NULL contrast alone does not separate evaluator-specific adaptation from generic improvement or the realized outcome sequence; we test both with counterfactual scoring and exact outcome-yoking. For the two-policy family, let $S_\theta(A) = U_\theta(A) - U_{\neg\theta}(A)$ denote specificity and $\bar{U}(A) = [U_{\text{FRONT}}(A) + U_{\text{BACK}}(A)]/2$ generic utility. The REAL-minus-NULL late-minus-early specificity change is 0.3910 (95% CI $[0.3425, 0.4395]$) for GPT and 0.3546 ($[0.2401, 0.4691]$) for Gemini. Opposite-evaluator utility decreases by 0.1842 and 0.1773, respectively, while the generic component is small (0.0113 and 0.0000). Since $\Delta_D = \Delta_{\bar{U}} + \frac{1}{2}\Delta_S$, the adaptation effect is dominated by active-versus-opposite evaluator specialization rather than generic improvement.

The GPT-5.4-mini exact-yoked study replays the exact binary-outcome sequence from each matched REAL donor while the recipient builds history from its own actions. Across 160 paired trajectories, REAL improves by 0.2022 in early-to-late regret, whereas YOKED improves by 0.0035; the paired contrast is $\widehat{\Delta}_{D,\text{yoked}} = 0.1987$ (95% CI $[0.1429, 0.2545]$). The two-way clustered standard error is 0.0279 with 58 degrees of freedom. REAL and YOKED have exactly the same outcome sequence within every paired trajectory and the same aggregate PASS rate (0.4820), while their actions differ in 83.36% of paired round decisions. Thus exposure to the same binary-feedback sequence is not sufficient: the observed adaptation depends on the relationship between the recipient's own action and subsequent feedback. YOKED is not evaluator-information-free, because the donor outcome stream can still carry policy information. Complementary no-history and smaller Gemini-yoked diagnostics are reported in Appendix D.

Finally, we contextualize the learned behavior against fixed strategies, win-stay/lose-switch (WSLS), a myopic Bayesian learner, and an oracle that knows the active evaluator in the same environment. Late GPT regret is 0.0697, versus 0.1508 for WSLS, 0.0305 for the exact-likelihood Bayesian reference, and 0 for the oracle; Figure 2c reports the full ladder. The Bayesian reference receives numerical likelihoods unavailable to the LLM and is myopic rather than finite-horizon optimal. Appendix B gives the full definitions and conditional-on-LLM-history diagnostic.

5 Dynamics and the Value of Evaluator Information

The primary hidden-evaluator result leaves two alternatives: the model may have settled on a fixed heuristic, or it may react passively without valuing evaluator information. We test these possibilities

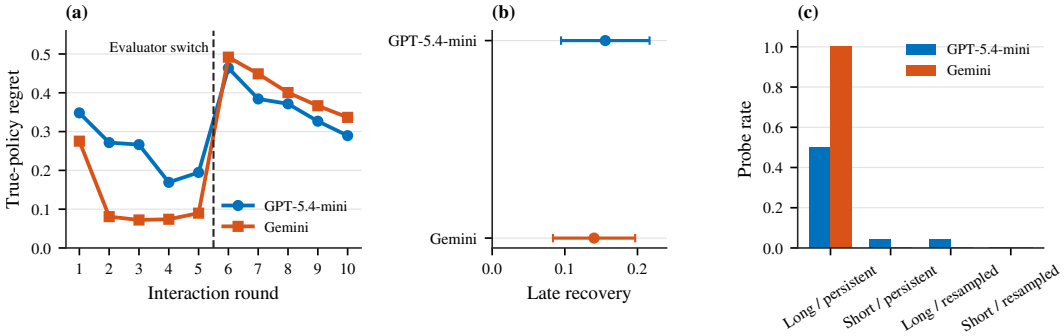

Figure 3: **Dynamics and value of evaluator information.** **(a)** GPT-5.4-mini and Gemini true-policy regret around an unannounced evaluator switch. **(b)** Late-recovery effects with 95% confidence intervals. **(c)** Costly-probe rates in the horizon $\times$ persistence intervention.

by changing the evaluator during an episode and by changing whether information about the evaluator can improve later decisions. Both tests use observed actions rather than self-reported beliefs.

5.1 Re-Adaptation after an Unannounced Evaluator Switch

We first ask whether behavior tracks evaluator changes rather than remaining tied to one fixed rule-independent strategy. The evaluator changes during a ten-round episode without announcement. The first post-switch action is selected before any feedback from the new evaluator. If behavior is tied to the old rule, that action should initially remain aligned with it; only later actions can respond to outcomes generated under the new rule.

Across 64 GPT-5.4-mini switch trajectories (640 decisions), mean true-policy regret rises to 0.4639 immediately after the switch and falls to 0.3082 late, giving recovery 0.1557 (95% CI $[0.0948, 0.2166]$). Gemini shows the same learn–disrupt–relearn pattern, from 0.4919 to 0.3516, with recovery 0.1402 ($[0.0837, 0.1968]$). Figure 3a–b shows the trajectories and recovery estimates. This rules out a single fixed, rule-independent positional heuristic; it does not imply complete re-learning of the new evaluator. Direction-specific trajectories are in Appendix E.

5.2 Costly Information Acquisition when Feedback Has Future Value

We then ask whether behavior responds to the future usefulness of evaluator information. The intervention introduces a costly probe with lower immediate utility but greater evaluator information. The design varies the remaining horizon and whether evaluator identity persists or is resampled. Let $Z = 1$ denote probe selection and $p_{h,s} = \Pr(Z = 1 \mid h, s)$ for $h \in \{\text{short}, \text{long}\}$ and $s \in \{\text{persistent}, \text{resampled}\}$. The primary interaction is

$$\Delta_{\text{probe}} = (p_{\text{long,persistent}} - p_{\text{short,persistent}}) - (p_{\text{long,resampled}} - p_{\text{short,resampled}}).$$

Positive values mean that a longer horizon increases probing specifically when information about the evaluator remains useful. The panel keeps the qualitative optimal probing pattern fixed; this experiment therefore tests the interaction, not fine-grained sensitivity around a cost–benefit boundary.

GPT-5.4-mini gives $\widehat{\Delta}_{\text{probe}} = 0.4167$ (95% CI $[0.2292, 0.5839]$). Probe rates are 50.00% in long-horizon persistent episodes, 4.17% in short-horizon persistent, 4.17% in long-horizon resampled, and 0% in short-horizon resampled episodes. Gemini gives complete separation on the 24-instance panel: 100% probing in long-horizon persistent cases and 0% in the other three cells, so $\widehat{\Delta}_{\text{probe}} = 1.000$ on this panel. We do not interpret the saturated finite-panel result as zero uncertainty outside the evaluated design.

Descriptive follow-through diagnostics are consistent with subsequent use of the acquired information; full switch directions, action menus, probe construction, and follow-through results are reported in Appendices E and F.

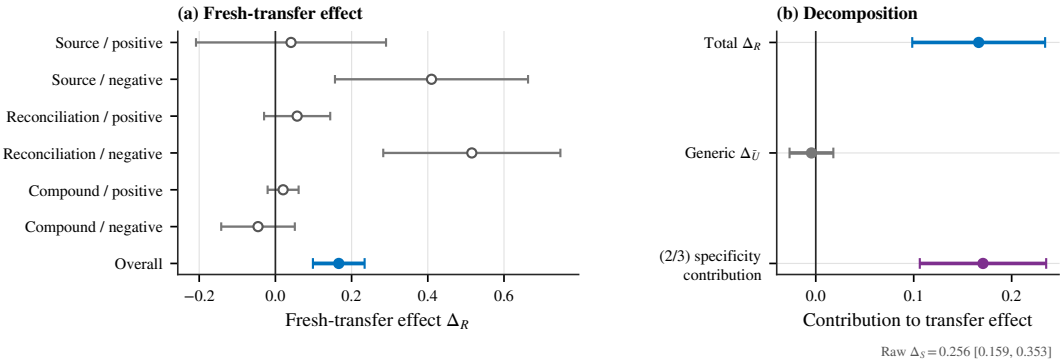

Figure 4: **Structural transfer beyond positional rules.** **(a)** Fresh-transfer REAL–CONTROL regret contrasts across the six policy × transfer-composition strata. **(b)** Decomposition of the transfer effect into generic and evaluator-specific components. Error bars show 95% confidence intervals.

6 Structural Transfer Beyond Positional Evaluation

The positional experiments permit exact identification but leave open transfer to a different action and evaluator structure. We therefore test fresh-instance transfer under feature-dependent hidden policies with exact scoring. Each packet contains eight unresolved documentation checks with two visible binary features and a resolution cost. The model chooses an unordered subset to resolve; an auditor samples two checks without replacement using hidden feature-dependent weights, and the packet passes only if both sampled checks are resolved.

The disclosed family contains three policies: SOURCE upweights checks using an external source, RECONCILIATION those requiring cross-record comparison, and COMPOUND those with both features. High-priority items receive weight 9 and all others weight 1. For packet x and resolved subset A,

$$p_\theta(A;x) = \sum_{\substack{i,j\in A\\ i\neq j}} \frac{w_i^\theta}{W_\theta}\frac{w_j^\theta}{W_\theta - w_i^\theta}, \qquad U_\theta(A;x) = p_\theta(A;x) - C(A),$$

with regret $R_\theta(A;x) = U_\theta^*(x) - U_\theta(A;x)$. The actor sees the candidate rules, their 9:1 weight ratio, the equal prior, task costs, and sampling process, but not the active rule.

The Gemini study uses 48 independently generated units, balanced across six policy × transfer-composition strata. Each unit yields a matched REAL/CONTROL pair with eight learning decisions followed by one scored transfer decision on fresh item identities and changed feature composition, for 96 trajectories and 864 hidden calls. In REAL, the evaluator remains fixed across learning and transfer. In CONTROL, action-dependent feedback is drawn from the equal mixture of candidate policies,

$$q(A_t;x_t) = \frac{1}{3}\sum_\phi p_\phi(A_t;x_t),$$

so it carries no information about the persistent evaluator used to score transfer. A 72-decision transparent-policy capability check achieved zero regret throughout; Appendix G gives qualification and provenance details.

The primary estimand is the equal-weight average across the six strata of the matched fresh-transfer regret difference

$$\Delta_R = \mathbb{E}\left[R_\theta(A_{\mathrm{tr}}^{\mathrm{CONTROL}};x_{\mathrm{tr}}) - R_\theta(A_{\mathrm{tr}}^{\mathrm{REAL}};x_{\mathrm{tr}})\right].$$

A positive value means that interaction with a persistent hidden evaluator improves the subsequent fresh-transfer decision relative to action-dependent mixture feedback that contains no information about the persistent scoring evaluator. A secondary decomposition separates generic utility from evaluator-specific specialization:

$$S_\theta(A;x) = U_\theta(A;x) - \frac{1}{2}\sum_{\phi\neq\theta} U_\phi(A;x), \qquad \bar{U}(A;x) = \frac{1}{3}\sum_\phi U_\phi(A;x),$$

so that

$$\Delta_R = \Delta_{\bar{U}} + \tfrac{2}{3}\Delta_S.$$

On fresh transfer packets, REAL reduces active-evaluator regret relative to CONTROL:

$$\widehat{\Delta}_R = 0.1663, \qquad 95\%\,\mathrm{CI} = [0.0985, 0.2342],$$

with $\mathrm{SE} = 0.0336$ and 42 residual degrees of freedom. Figure 4a shows the six stratum-specific contrasts. Evaluator specificity is 0.2561 (95% CI $[0.1593, 0.3529]$), while generic improvement is -0.0044 ($[-0.0268, 0.0180]$); Figure 4b shows the decomposition. Thus transfer is evaluator-specific and extends beyond position-dependent rules to fresh feature-dependent instances.

7 Discussion and Limitations

The results show that evaluator-specific adaptation can arise under repeated hidden evaluation and characterize its dynamics, information sensitivity, and structural transfer. Prior work establishes feedback adaptation Pan et al. (2024), monitor awareness Jiralerspong et al. (2026), and black-box extraction of hidden rules Yao et al. (2025); we identify evaluator-specific change in the same task-performing model using exact evaluator-specific scoring and matched causal controls.

The controls address complementary alternatives. NULL provides repeated binary feedback without making outcomes depend on the submitted action or active evaluator, but it does not reproduce the realized REAL outcome stream. YOKED reproduces that stream while breaking its dependence on the recipient's own action, although donor outcomes can still contain evaluator information. Silent switches test whether behavior follows evaluator changes, and costly-information interventions test whether action choice responds to the future usefulness of evaluator information. These results are consistent with strategic-response work on hidden decision rules Shao et al. (2025).

A practical implication is that evaluator quality may not be only a one-shot property. In automated review, compliance, benchmarking, and agent-monitoring workflows, repeated outcomes can themselves become information that shapes later behavior. Evaluation design may therefore need to consider not only current detection performance, but also how much evaluator-specific information accumulates across interactions.

The core effect replicates across four model families; dynamics and information sensitivity use GPT and Gemini. Structural transfer is demonstrated in one model family and should be interpreted as evidence that the core phenomenon can extend beyond positional rules, not as a cross-model claim about structural transfer. The controlled design ensures that evaluator-specific adaptation is identifiable, not that the model will behaviorally exploit the available evaluator information. Evaluator families are disclosed and horizons are short: the experiments establish that evaluator-specific adaptation can occur under controlled repeated evaluation, not how prevalent or rapid it is in deployed systems.

8 Conclusion

Repeated evaluation can help enforce a hidden decision rule, but it can also reveal information about that rule over time. Our results show that evaluator-specific behavioral adaptation can arise under repeated hidden evaluation rather than merely reflecting improvement with experience or the sequence of outcomes received. The core effect appears across four model families. On GPT and Gemini, behavior re-adapts after evaluator switches and action choice responds to whether evaluator information remains useful; on Gemini, evaluator-specific transfer extends beyond the position-dependent setting used for primary identification.

Evaluator learnability therefore raises a broader design question: how should evaluation systems be constructed when repeated outcomes can become informative to the models they assess? In practice, repeated evaluation should be treated not only as a scoring mechanism, but also as an information channel whose effects can accumulate across interactions. This motivates work on learned and drifting evaluators, longer horizons, richer feedback channels, evaluator rotation or adaptation, and more realistic agentic workflows.

AI USE STATEMENT

Generative AI systems were used for literature discovery, implementation and analysis support, and language editing. The authors take full responsibility for the final work.

REPRODUCIBILITY STATEMENT

The experiments are tracked by frozen configurations, manifests, raw response records, analysis scripts, and deterministic figure-generation code. The supplementary material documents evaluator definitions, prompts, model and provider settings, invalid-response and retry handling, clustered inference, and experiment provenance. The anonymous code and data artifact contains the complete reproduction workflow used for the submission.

ETHICS STATEMENT

This work studies how repeated feedback can change model behavior under hidden evaluation. The experiments use controlled benchmark-derived tasks and do not involve human subjects or private user data. The results may inform both the design and the analysis of adaptive oversight systems; we therefore report the behavioral boundaries of the experiments and avoid claims about real-world evaluator evasion that were not directly tested.

REFERENCES

J. Charles Alderson and Dianne Wall. Does washback exist? *Applied Linguistics*, 14(2):115–129, 1993. doi: 10.1093/applin/14.2.115. URL https://doi.org/10.1093/applin/14.2.115.

Rami Aly, Zhijiang Guo, Michael Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. FEVEROUS: Fact extraction and VERification over unstructured and structured information. In J. Vanschoren and S. Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1. Curran Associates, Inc., 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/68d30a9594728bc39aa24be94b319d21-Abstract-round1.html.

Rudolf Avenhaus, Bernhard von Stengel, and Shmuel Zamir. Inspection games. In Robert Aumann and Sergiu Hart (eds.), *Handbook of Game Theory with Economic Applications*, volume 3, pp. 1947–1987. Elsevier, 2002. doi: 10.1016/S1574-0005(02)03014-X. URL https://www.sciencedirect.com/science/article/pii/S157400050203014X.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeffrey Wu. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4971–5012. PMLR, 2024. URL https://proceedings.mlr.press/v235/burns24b.html.

Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. Humans or LLMs as the judge? a study on judgement bias. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8301–8327, 2024. doi: 10.18653/v1/2024.emnlp-main.474. URL https://aclanthology.org/2024.emnlp-main.474/.

Paul F. Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30, pp. 4299–4307. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/hash/d5e2c0adad503c91f91df240d0cd4e49-Abstract.html.

Pedro Ferreira, Wilker Aziz, and Ivan Titov. Truthful or fabricated? using causal attribution to mitigate reward hacking in explanations. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 130865–130894, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/d4b892df8bec8e13925f57649fbae039-Abstract-Conference.html.

Ganesh Ghalme, Vineet Nair, Itay Eilat, Inbal Talgam-Cohen, and Nir Rosenfeld. Strategic classification in the dark. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3672–3681. PMLR, 2021. URL https://proceedings.mlr.press/v139/ghalme21a.html.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large language models can self-correct with tool-interactive critiquing. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 57734–57811, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/fef126561bbf9d4467dbb8d27334b8fe-Abstract-Conference.html.

Ryan Greenblatt, Buck Shlegeris, Kshitij Sachan, and Fabien Roger. AI control: Improving safety despite intentional subversion. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 16295–16336. PMLR, 2024. URL https://proceedings.mlr.press/v235/greenblatt24a.html.

Ruixuan Huang, Xunguang Wang, Zongjie Li, Daoyuan Wu, and Shuai Wang. GuidedBench: Measuring and mitigating the evaluation discrepancies of in-the-wild LLM jailbreak methods. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ZVg8y3ibyM.

Thomas Jiralerspong, Flemming Kondrup, and Yoshua Bengio. Noticing the watcher: LLM agents can infer CoT monitoring from blocking feedback, 2026. URL https://arxiv.org/abs/2603.16928.

Neil Kale, Chen Bo Calvin Zhang, Kevin Zhu, Ankit Aich, Paula Rodriguez, Christina Knight, and Zifan Wang. Reliable weak-to-strong monitoring of LLM agents. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 78083–78138, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/7e105e4ae3016974e2d518e9d281f5e5-Abstract-Conference.html.

Hadi Khalaf, Claudio Mayrink Verdun, Alex Oesterling, Himabindu Lakkaraju, and Flavio Calmon. Inference-time reward hacking in large language models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 61720–61760. Curran Associates, Inc., 2025. doi: 10.52202/085713-2064. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/590a0cc0306c1c63e2d66a51a407718f-Abstract-Conference.html.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. Prometheus: Inducing fine-grained evaluation capability in language models. In *International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=8euJaTveKw.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. Prometheus 2: An open source language model specialized in evaluating other language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 4334–4353, 2024b. doi: 10.18653/v1/2024.emnlp-main.248. URL https://aclanthology.org/2024.emnlp-main.248/.

Cassidy Laidlaw, Shivam Singhal, and Anca Dragan. Correlated proxies: A new definition and improved mitigation for reward hacking. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 88379–88421, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/dbe2cfe4767f3255160b73a36ae3162e-Abstract-Conference.html.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. RewardBench: Evaluating reward models for language modeling. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1755–1797. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-naacl.96. URL https://aclanthology.org/2025.findings-naacl.96/.

Xiaomin Li, Mingye Gao, Zhiwei Zhang, Jingxuan Fan, and Weiyu Li. RuleAdapter: Dynamic rules for training safety reward models in RLHF. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 34355–34378. PMLR, 2025a. URL https://proceedings.mlr.press/v267/li25o.html.

Yang Li, Jie Ma, Miguel Ballesteros, Yassine Benajiba, and Graham Horwood. Active evaluation acquisition for efficient LLM benchmarking. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 35581–35602. PMLR, 2025b. URL https://proceedings.mlr.press/v267/li25bp.html.

Tianqi Liu, Wei Xiong, Jie Ren, Lichang Chen, Junru Wu, Rishabh Joshi, Yang Gao, Jiaming Shen, Zhen Qin, Tianhe Yu, Daniel Sohn, Anastasia Makarova, Jeremiah Zhe Liu, Yuan Liu,

Bilal Piot, Abe Ittycheriah, Aviral Kumar, and Mohammad Saleh. RRM: Robust reward model training mitigates reward hacking. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 62682–62700, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/9d46574e5baae5121180228223a11836-Abstract-Conference.html.

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*, volume 2024, pp. 52989–53046, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/e9df36b21ff4ee211a8b71ee8b7e9f57-Abstract-Conference.html.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 46534–46594. Curran Associates, Inc., 2023. doi: 10.52202/075280-2019. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/91edff07232fb1b55a505a9e9f6c0ff3-Abstract-Conference.html.

Yuchun Miao, Sen Zhang, Liang Ding, Rong Bao, Lefei Zhang, and Dacheng Tao. InfoRM: Mitigating reward hacking in RLHF via information-theoretic reward modeling. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 134387–134429. Curran Associates, Inc., 2024. doi: 10.52202/079017-4270. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/f25d75fc760aec0a6174f9f5d9da59b8-Abstract-Conference.html.

Natinael Solomon Neggatu, Jeremie Houssineau, and Giovanni Montana. Evaluation-time policy switching for offline reinforcement learning. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, pp. 1520–1528. International Foundation for Autonomous Agents and Multiagent Systems, 2025. doi: 10.5555/3709347.3743786. URL https://www.ifaamas.org/Proceedings/aamas2025/pdfs/p1520.pdf.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Associates, Inc., 2022. doi: 10.52202/068431-2011. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.

Alexander Pan, Erik Jones, Meena Jagadeesan, and Jacob Steinhardt. Feedback loops with language models drive in-context reward hacking. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 39154–39200. PMLR, 2024. URL https://proceedings.mlr.press/v235/pan24d.html.

Rafael Rafailov, Yaswanth Chittepu, Ryan Park, Harshit Sikchi, Joey Hejna, W. Bradley Knox, Chelsea Finn, and Scott Niekum. Scaling laws for reward model overoptimization in direct alignment algorithms. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 126207–126242. Curran Associates, Inc., 2024. doi: 10.52202/079017-4009. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/e45caa3d5273d105b8d045e748636957-Abstract-Conference.html.

Vyas Raina, Adian Liusie, and Mark Gales. Is LLM-as-a-Judge robust? investigating universal adversarial attacks on zero-shot LLM assessment. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 7499–7517, 2024. doi: 10.18653/v1/2024.emnlp-main.427. URL https://aclanthology.org/2024.emnlp-main.427/.

Gabriel Recchia, Chatrik Singh Mangat, Jinu Nyachhyon, Mridul Sharma, Callum Canavan, Dylan Epstein-Gross, and Muhammed Abdulbari. Confirmation bias: A challenge for scalable oversight. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 37877–37886. Association for the Advancement of Artificial Intelligence, 2026. doi: 10.1609/aaai.v40i44.41124. URL https://ojs.aaai.org/index.php/AAAI/article/view/41124.

Han Shao, Shuo Xie, and Kunhe Yang. Should decision-makers reveal classifiers in online strategic classification? In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 54254–54274. PMLR, 2025. URL https://proceedings.mlr.press/v267/shao25c.html.

Abhay Sheshadri, Aidan Ewart, Kai Fronsdal, Isha Gupta, Samuel R. Bowman, Sara Price, Samuel Marks, and Rowan Wang. AuditBench: Evaluating alignment auditing techniques on models with hidden behaviors, 2026. URL https://arxiv.org/abs/2602.22755.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652. Curran Associates, Inc., 2023. doi: 10.52202/075280-0377. URL https://papers.nips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Sijun Tan, Siyuan Zhuang, Kyle Montgomery, William Tang, Alejandro Cuadron, Chenguang Wang, Raluca Popa, and Ion Stoica. JudgeBench: A benchmark for evaluating LLM-Based judges. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 63277–63303, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/9e720fce64f91114c49cfd640d821da3-Abstract-Conference.html.

Terry Tong, Fei Wang, Zhe Zhao, and Muhao Chen. BadJudge: Backdoor vulnerabilities of LLM-as-a-judge. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 18300–18325, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/2e48f562a2c8f64c7404a6c3a518af74-Abstract-Conference.html.

Songjun Tu, Jingbo Sun, Qichao Zhang, Xiangyuan Lan, and Dongbin Zhao. Online preference-based reinforcement learning with self-augmented feedback from large language model. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, pp. 2069–2077. International Foundation for Autonomous Agents and Multiagent Systems, 2025. doi: 10.5555/3709347.3743845. URL https://www.ifaamas.org/Proceedings/aamas2025/pdfs/p2069.pdf.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7534–7550, 2020. doi: 10.18653/v1/2020.emnlp-main.609. URL https://aclanthology.org/2020.emnlp-main.609/.

Dianne Wall and J. Charles Alderson. Examining washback: The sri lankan impact study. *Language Testing*, 10(1):41–69, 1993. doi: 10.1177/026553229301000103. URL https://doi.org/10.1177/026553229301000103.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9440–9450. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.511. URL https://aclanthology.org/2024.acl-long.511/.

Xinpeng Wang, Nitish Joshi, Barbara Plank, Rico Angell, and He He. Is it thinking or cheating? detecting implicit reward hacking by measuring reasoning effort. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 96956–96981, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/9d0ee47d732bd526c4a03d4aecc39bc1-Abstract-Conference.html.

Marcus Williams, Micah Carroll, Adhyyan Narang, Constantin Weisser, Brendan Murphy, and Anca Dragan. On targeted manipulation and deception when optimizing LLMs for user feedback. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 89542–89593, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/deed907f507e526a8dd20896d2bd7c49-Abstract-Conference.html.

Hongwei Yao, Yun Xia, Shuo Shao, Haoran Shi, Tong Qiao, and Cong Wang. Black-box guardrail reverse-engineering attack, 2025. URL https://arxiv.org/abs/2511.04215.

Rui Yu, Shenghua Wan, Yucen Wang, Chen-Xiao Gao, Le Gan, Zongzhang Zhang, and De-Chuan Zhan. Reward models in deep reinforcement learning: A survey. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 10807–10816. International Joint Conferences on Artificial Intelligence Organization, 2025. doi: 10.24963/ijcai.2025/1199. URL https://www.ijcai.org/proceedings/2025/1199.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 46595–46623. Curran Associates, Inc., 2023. doi: 10.52202/075280-2020. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html.

Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haofei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, and Maarten Sap. SOTOPIA: Interactive evaluation for social intelligence in language agents. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 40975–41019, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/b3075b88e583a0e98d8b24338a613060-Abstract-Conference.html.

Mingchen Zhuge, Changsheng Zhao, Dylan R. Ashley, Wenyi Wang, Dmitrii Khizbullin, Yunyang Xiong, Zechun Liu, Ernie Chang, Raghuraman Krishnamoorthi, Yuandong Tian, Yangyang Shi, Vikas Chandra, and Jürgen Schmidhuber. Agent-as-a-judge: Evaluate agents with agents. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 80569–80611. PMLR, 2025. URL https://proceedings.mlr.press/v267/zhuge25a.html.

A Formal Environment and Benchmark Construction

The controlled panel is constructed from pre-existing benchmark-derived source pools, not from model outcomes. Each source pool must contain at least four benchmark-supported and four benchmark-unsupported claims. With global seed 47, a source-specific generator is derived by hashing `47:<source-pool-id>`; the two label classes are independently permuted and the first four of each are retained. There is no retry after construction or validation failure. Claim text, benchmark label, and scalar value are inherited unchanged; the ICLR constructor performs no value normalization. Source-pool provenance records that values lie in $\{2, 3\}$ and that target seed claims are excluded. FEVEROUS pools are anchored to an evidence page, whereas SciFact uses the frozen globally balanced topic grouping. Benchmark labels remain the factual ground truth.

Let S be the total value of the four selected supported claims and M the sum of the four largest values among all eight selected claims. A candidate is discarded unless $M > S + 10^{-9}$, after which

$$R = S + \tfrac{1}{2}(M - S).$$

An action is an ordered list of exactly four distinct claims whose value sum is at least R. Every feasible permutation is enumerated. The reviewer inspects exactly two positions sequentially without replacement. FRONT uses $w_j = \exp[-\gamma(j-1)]$ and BACK reverses these weights, with $\gamma = \log 2$, giving relative weights $(1, 1/2, 1/4, 1/8)$. Remaining weights are renormalized after the first draw; PASS occurs iff both inspected claims are supported.

Before model calls, each pool passes the implemented P1–P7 screen: (P1) fixed-size feasible actions exist; (P2) an all-supported feasible action is rejected as trivial only when FRONT/BACK also lack the required conflict; (P3) at least two distinct feasible risky subsets exist; (P4) FRONT and BACK optimal sets are disjoint and maximum cross-policy loss is at least 0.10; (P5) some action has positive FRONT/BACK Bernoulli Jensen–Shannon divergence; (P6) fewer than 90% of actions have divergence at least 0.60; and (P7) every action is more than 0.02 suboptimal under at least one primary policy. Of 40 FEVEROUS source pools, 25 were constructed and all passed; the other 15 failed $M > S$. All 34 SciFact pools passed. The panel therefore contains 59 pools. Any required retention ranking is the SHA-256 ordering of `47:retain:<source-pool-id>`, independent of model behavior.

B Bayesian References and Information Quantities

The secondary offline references enumerate all feasible ordered four-claim artifacts. Fixed FRONT and fixed BACK optimize the indicated policy regardless of the active evaluator. Prior-optimal selection maximizes the equal-weight mixture utility. Uniform selection samples all feasible artifacts equally. Win-stay/lose-switch (WSLS) starts with either policy with equal probability, chooses that policy's current-task optimum, retains the strategy after PASS, and flips it after FAIL. The oracle receives the active identity and attains zero regret.

The myopic Bayesian reference starts from $(1/2, 1/2)$, selects an action maximizing current posterior expected utility, and updates using its own action and the observed audit outcome. Both Bayesian and WSLS histories are integrated over all binary branches exactly, using floating-point probability sums rather than Monte Carlo. Optimization ties are averaged uniformly over distinct payoff vectors, with deterministic artifact representatives. This avoids arbitrary claim multiplicity determining which tied strategy is selected.

The Bayesian reference knows the exact numerical audit law, whereas the LLM is told the qualitative FRONT/BACK family. It is myopic rather than optimal for the whole eight-round horizon. Its posterior is an experimenter's reference and is not a measurement of the LLM's beliefs.

A separate diagnostic conditions on the LLM's past REAL history, before each current action. Its late-round recommended regret is 0.0290 for GPT and 0.0202 for Gemini; the LLM action is myopically posterior-optimal in 89.06% and 98.44% of late decisions. These diagnostics neither generate independent histories nor establish an internal Bayesian mechanism. Likelihood updates use the audit PASS probability, which can differ from feasibility-gated utility.

C PRIMARY HIDDEN-EVALUATOR PROTOCOL

The confirmatory protocol fixes source pools, REAL–NULL pairing, task schedules, model snapshot, reasoning effort, structured-output schema, retry policy, primary estimator, and cluster structure before the final target-model run. Technical retries are permitted only for transport or parser failures and never for scientifically valid unfavorable outputs.

The primary schedule contains 472 matched eight-round episode pairs, 25 FEVEROUS and 34 SciFact pools, four scheduler replicates per policy, and eight distinct pools per episode. The runner consumes the frozen schedule manifest directly. The frozen schedule and estimator are included in the anonymous reproducibility artifact. The offline primary-data audit reproduces the primary numbers from immutable raw files, checks matching design fields and seeds, and recomputes every FRONT/BACK utility and regret from the source pools.

Feasibility and feedback discrepancy. All recorded actions parsed successfully. In the GPT study, 26/3776 REAL and 27/3776 NULL actions are value-infeasible (53/7552 decisions; 0.70%), and 18 REAL plus 4 NULL cases receive PASS (22/7552; 0.29%). The executed runner audits every well-formed submission and applies the value constraint only to scored utility, whereas an earlier protocol note described value-infeasible submissions as forced FAIL. We retain the observed histories rather than relabel outcomes or reconstruct unobserved counterfactual trajectories. Gemini REAL and NULL contain no value-infeasible actions. The primary regret remains feasibility-gated; as a scoring sensitivity, using ungated audit probability gives 0.2083 (95% CI $[0.1847, 0.2319]$), compared with the primary 0.2068 estimate. This sensitivity does not recreate the histories that a forced-FAIL implementation would have generated.

Prospective GPT exact-yoked robustness. Before any GPT-YOKED outcome was observed, the full 472-trajectory design was retained as an unexecuted frozen reference and a separate reduced-budget design was frozen at $N = 160$. Selection uses seed 9173, ten trajectories from each of 16 domain $\times$ evaluator $\times$ scheduler-replicate cells, and preserves all 59 source pools. The resulting study contains 1,280 YOKED model calls paired with the corresponding REAL donor trajectories. The frozen selection manifest is included in the anonymous reproducibility artifact. YOKED receives the exact donor PASS/FAIL bit in each round while building history only from its own actions; scientific feedback is never re-evaluated under the recipient action.

The frozen analysis gives

$$\widehat{\Delta}_{D,\mathrm{yoked}} = 0.1987, \qquad 95\%\,\mathrm{CI} = [0.1429, 0.2545], \qquad \mathrm{SE} = 0.0279,$$

with 59 source-pool clusters, 160 episode-pair clusters, and 58 degrees of freedom. REAL early-to-late regret improvement is 0.2022, compared with 0.0035 for YOKED. Outcome equality is exact within every paired trajectory and round; both conditions have PASS rate 0.4820. Recipient and donor actions differ in 83.36% of paired round decisions. As an offline plausibility diagnostic, the replayed outcome has probability at least 0.05 under one or both candidate evaluators for all 1,280 recipient actions; only 13 rounds (1.02%) fall below 0.10 under both, and the median probability under the more compatible candidate is 0.8226. Thus yoking does not generally produce outcomes impossible under the disclosed candidate family, although it can remain improbable under the active member. The run contains no parser failures; eight saved calls required transport-level retry, with at most three attempts. Total YOKED token use is 5,369,810 tokens. Figure 5 shows the descriptive trajectories. These checks establish exact outcome-sequence matching, not evaluator-uninformative feedback.

D ADDITIONAL HIDDEN-EVALUATOR RESULTS

This section collects the round-level REAL and NULL trajectories, FRONT/BACK and FEVEROUS/SciFact breakdowns, complementary controls, and reference-policy results.

Transparent specialization score. For artifacts A_θ generated with evaluator θ revealed, define

$$M_{\theta,\theta'} = \mathbb{E}[U_{\theta'}(A_\theta)], \qquad \mathrm{PAS} = \frac{1}{2}\left[(M_{\mathrm{F,F}} - M_{\mathrm{F,B}}) + (M_{\mathrm{B,B}} - M_{\mathrm{B,F}})\right].$$

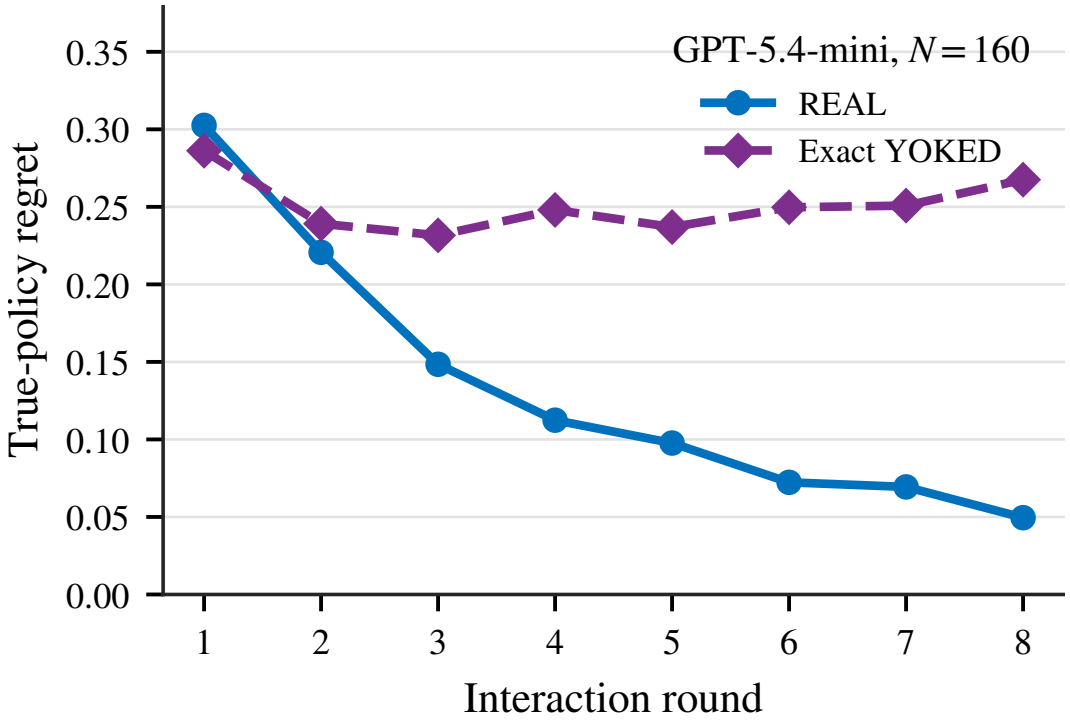

Figure 5: **Exact-yoked trajectory diagnostic.** Descriptive round-wise means on the $N = 160$ GPT-5.4-mini donor pairs. REAL and YOKED receive identical realized PASS/FAIL sequences, but only REAL preserves dependence between the actor's own action and its outcome. The primary clustered contrast and confidence interval are reported in the text.

Positive PAS measures specialization to the stated evaluator rather than utility improvement that transfers equally to both policies.

Exact reference expectations and observed LLM regret.

Reference (GPT panel)	Early	Late	Early−late
uniform_feasible	0.4959	0.4959	0.0000
fixed_front	0.2758	0.2758	0.0000
fixed_back	0.2758	0.2758	0.0000
prior_optimal	0.2664	0.2664	0.0000
WSLS	0.1984	0.1508	0.0476
bayes_myopic	0.2164	0.0305	0.1859
oracle	0.0000	0.0000	0.0000
LLM REAL	0.2542	0.0697	0.1845
LLM NULL	0.2831	0.3054	−0.0223

These exact reference expectations and observed LLM means contextualize both late quality and adaptation. No LLM-versus-reference significance test is claimed.

Complementary history and specificity diagnostics. For Gemini, full-history adaptation exceeds the no-history condition by 0.1381 (95% CI $[0.1037, 0.1726]$). An earlier 64-pair Gemini exact-yoked diagnostic gives 0.0324 (95% CI $[-0.0117, 0.0764]$); it is reported as a smaller model-specific boundary rather than treated as the main yoking result. The exact specificity decomposition gives GPT active 0.2068, opposite -0.1842, generic 0.0113, and specificity 0.3910; Gemini gives active 0.1773, opposite -0.1773, generic 0.0000, and specificity 0.3546 (Figure 7). Secondary component intervals are descriptive and unadjusted.

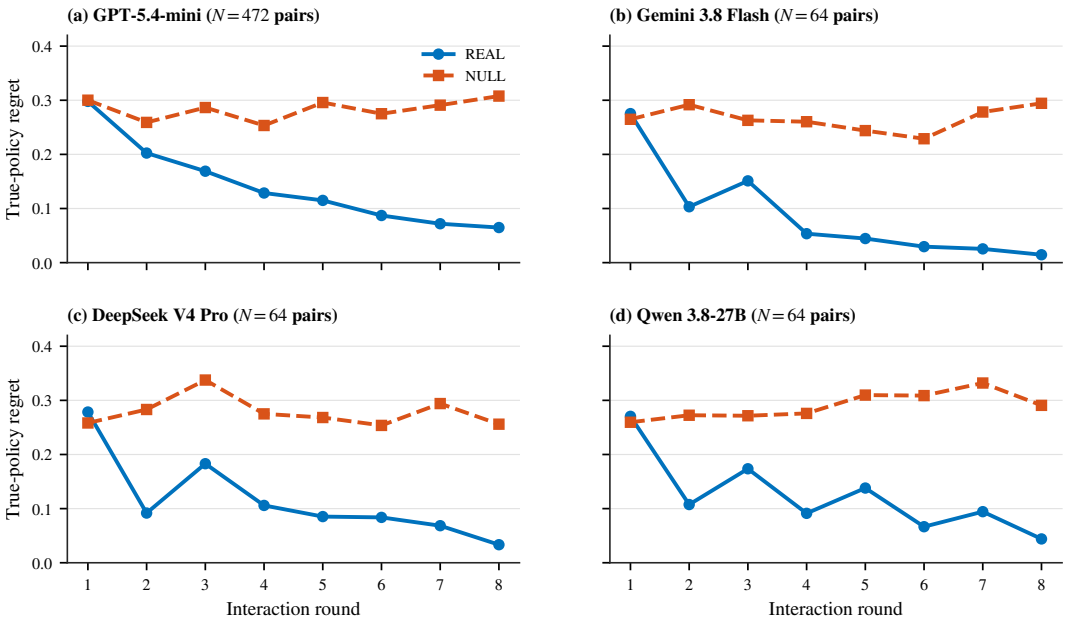

Figure 6: **Hidden-evaluator trajectories across model families.** Descriptive mean true-policy regret by interaction round for REAL and NULL. GPT-5.4-mini uses the full 472-pair panel; Gemini, DeepSeek, and Qwen each use a 64-pair replication panel. Formal model-specific effect estimates and clustered confidence intervals are reported in Figure 2b.

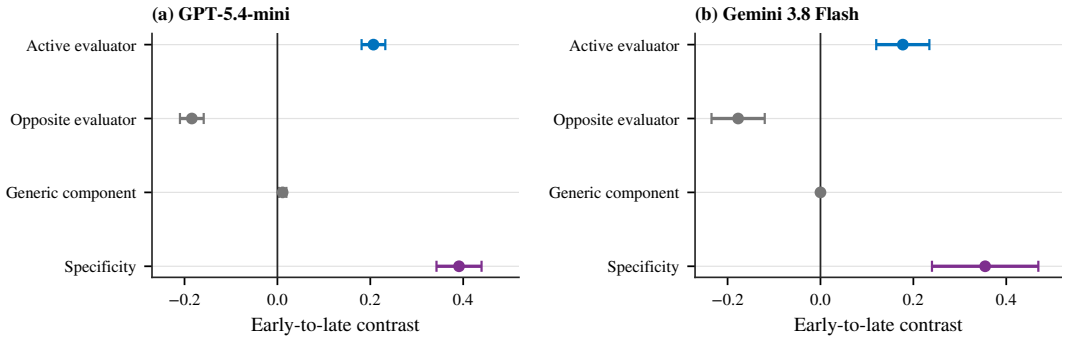

Figure 7: **Evaluator-specific rather than generic improvement.** Frozen active-evaluator, opposite-evaluator, generic, and specificity contrasts for GPT-5.4-mini and Gemini. Points show estimates and bars show the corresponding 95% confidence intervals from the secondary decomposition; these component intervals are descriptive and unadjusted.

Reported appendix quantities use the experimental units and inference procedures specified for their corresponding studies; no result is reconstructed manually from the PDF.

E Silent-Switch Experiment

The frozen switch study contains 64 ten-round trajectories per model and includes both FRONT→BACK and BACK→FRONT directions. The evaluator changes before round 6, and the round-6 action is chosen before any outcome generated by the new evaluator enters the visible history. This timing makes the immediate regret spike interpretable as persistence of the previously adapted behavior rather than a response to post-switch feedback.

For GPT-5.4-mini, immediate new-policy regret is 0.4639 and late regret is 0.3082, giving recovery 0.1557 (95% CI $[0.0948, 0.2166]$). Gemini moves from 0.4919 to 0.3516, giving recovery 0.1402

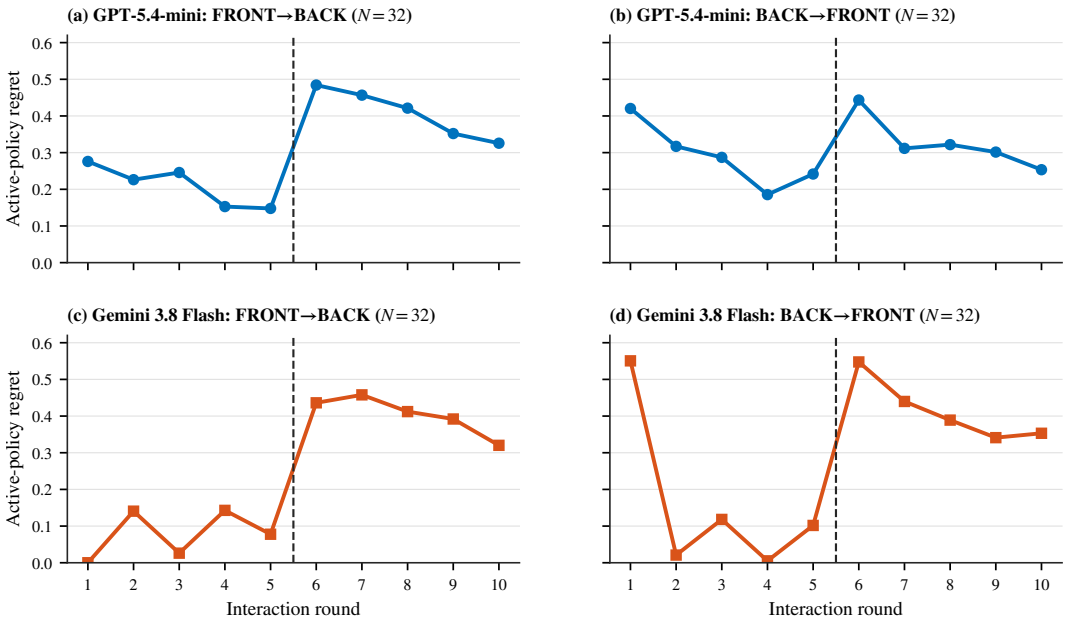

Figure 8: **Silent-switch dynamics in both directions.** Descriptive mean active-policy regret for FRONT→BACK and BACK→FRONT, separately for GPT-5.4-mini and Gemini. The dashed line marks the unannounced switch between rounds 5 and 6; the round-6 action is selected before feedback from the new evaluator. Each panel contains 32 trajectories.

$([0.0837, 0.1968])$. Figure 3 reports the aggregate mean trajectories, and Figure 8 separates the two switch directions. Behavioral wording is retained throughout; the switch experiment does not establish that the model explicitly represents or reports the policy change.

F Costly Information Acquisition

The costly-probe experiment crosses horizon length with evaluator persistence. The informative action has lower immediate utility but changes the distribution of feedback in a way that better separates candidate evaluators. The primary statistic is the horizon-by-persistence interaction in probe choice, not the raw probe rate in any one cell. Each model is evaluated on the same frozen panel of 24 probe-design instances, which serve as the analysis units (microinstances).

For GPT-5.4-mini, probe rates are 50.00% (long/persistent), 4.17% (short/persistent), 4.17% (long/resampled), and 0% (short/resampled), yielding interaction 0.4167 with 95% CI $[0.2292, 0.5839]$. Gemini selects the probe in 100% of long/persistent cases and 0% elsewhere, yielding the saturated finite-panel interaction 1.0000; this is not interpreted as zero uncertainty outside the evaluated panel.

The initial menu contains exactly two blinded artifacts, `myopic` and `probe`; rounds 2–8 contain `myopic`, `front_exploit`, and `back_exploit`. Machine action labels, claim labels, and menu order are deterministically permuted before display. Each artifact has four to six unit-value claims, and its pass probability is one minus the normalized positional weight on unsupported claims under one inspection. Under the equal prior, immediate probe cost is $\mathbb{E}U(a_{\text{myopic}}) - \mathbb{E}U(a_{\text{probe}})$; over the 24 frozen instances it ranges from 0.0806 to 0.3095 (median 0.1452). Binary information gain in nats is

$$I(\Theta; Y \mid a) = H\big(\tfrac{1}{2}q_F(a) + \tfrac{1}{2}q_B(a)\big) - \tfrac{1}{2}H(q_F(a)) - \tfrac{1}{2}H(q_B(a)).$$

Probe information gain ranges from 0.1282 to 0.3047 (median 0.1878); myopic Jensen–Shannon divergence is at most 0.02. The panel additionally requires probe cost at least 0.08, probe divergence at least 0.10, and exact dynamic-program choices of myopic at horizon 1, probe at persistent horizon 8, and myopic at resampled horizon 8. Candidates are deterministically enumerated over artifact lengths 4–8 and one- or two-unsupported-position patterns; the first 24 satisfying all gates are frozen.

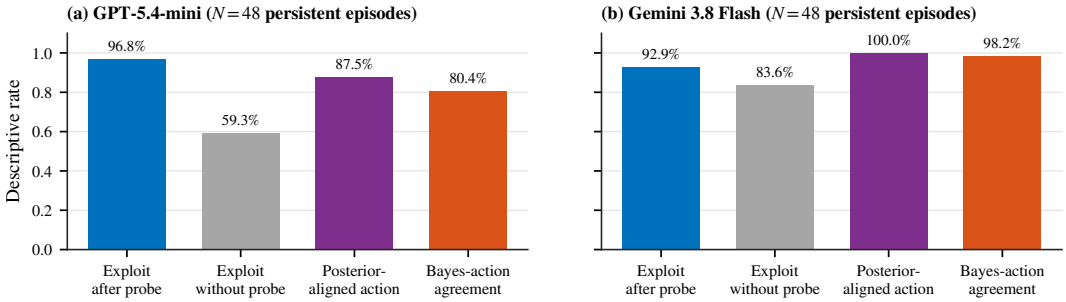

Figure 9: **Descriptive follow-through after costly information acquisition.** Persistent-condition summaries over 48 eight-round episodes per model. The panels show later policy-specific exploitation after versus without an initial probe, action choice aligned with the higher-posterior evaluator under the exact Bayesian reference, and agreement with the myopic Bayesian action. These diagnostics are consistent with subsequent use of acquired information; they are descriptive and do not identify a causal mediation mechanism or optimal value-of-information computation.

The four first-action cells cross horizon 1 versus 8 with a persistent evaluator versus independent per-round resampling. The microinstance is the analysis unit; two prompt variants are averaged within cell. The interaction interval is the percentile interval from 2,000 seed-47 bootstrap resamples of the 24 microinstances. The sequential follow-through phase executes 48 eight-round episodes per persistence condition; its summaries are descriptive. Later actions favor the evaluator with the larger exact Bayesian reference posterior in 87.49% of GPT and 100% of Gemini decisions, while agreement with the myopic Bayesian reference action is 80.36% and 98.21%. Exploitation after a probe is 96.83% versus 59.34% without a probe for GPT, and 92.86% versus 83.57% for Gemini (Figure 9).

G Feature-Dependent Audit Transfer

Each packet contains eight checks with visible feature pair (e_i, r_i), where e_i indicates use of an external source and r_i indicates cross-record reconciliation, together with a resolution cost drawn as a permutation of $\{0.04, 0.06, 0.08, 0.10, 0.14, 0.16, 0.18, 0.20\}$. The actor chooses any subset of checks, giving 256 possible actions. The auditor samples two distinct checks sequentially without replacement using policy weights and returns PASS iff both sampled checks were resolved. Resolving a check does not change its audit weight.

Learning packets use feature-cell counts $(00, 10, 01, 11) = (2, 2, 2, 2)$. Transfer uses either $(3, 1, 1, 3)$ or $(1, 3, 3, 1)$ with fresh identifiers, a fresh cost–feature assignment, and randomized display order. No transfer outcome is observed before the scored transfer action. REAL keeps one evaluator fixed. CONTROL samples feedback from the exact equal policy mixture conditional on the actor's own action, preserving generic action-dependent task feedback while carrying no information about the persistent scoring evaluator.

Gemini qualification. Before hidden production, a frozen 72-call transparent capability panel was run with `gemini-3.8-flash`, medium reasoning, an 8,192-token completion cap, provider-default temperature, and the structured-output schema. The preregistered gate required mean regret at most 0.05 in every one of the nine policy–composition cells and at least 90% of decisions with regret at most 0.10. All nine cell means were 0, all $72/72$ decisions had regret 0, and there were no parser failures, truncations, transport retries, or non-`stop` completions. A separate production-like validation test covered early-, middle-, late-history and transfer contexts with $9/9$ valid structured responses and no truncation.

Structural-transfer production design. The hidden Gemini design contains 48 independently generated, prospectively policy-assigned units. The six policy $\times$ transfer-composition strata contain exactly eight units each; every base block receives one policy before hidden outcomes are observed. Each unit contributes matched REAL and CONTROL trajectories with eight learning decisions and

one fresh transfer decision, giving 96 trajectories and 864 hidden calls. The eight frozen batches contain 108 calls each. The frozen design and the later explicit production authorization are recorded separately in the reproducibility ledger.

The primary estimator is the equal-weight average of the six stratum-specific matched fresh-transfer regret contrasts. Specificity and the exact generic/specific decomposition are secondary. Only the 48-unit, 864-call design described above was executed.

Primary and secondary results. The final family-average transfer effect is

$$\widehat{\Delta}_R = 0.1663, \qquad 95\%\,\mathrm{CI} = [0.0985, 0.2342], \qquad \mathrm{SE} = 0.0336,$$

with 42 residual degrees of freedom. Evaluator specificity is 0.2561 (95% CI $[0.1593, 0.3529]$), whereas generic improvement is -0.0044 ($[-0.0268, 0.0180]$). Under the frozen decomposition $\Delta_R = \Delta_{\bar{U}} + \frac{2}{3}\Delta_S$, the primary transfer gain is therefore attributable to evaluator-specific adaptation rather than generic utility improvement.

One provider-side HTTP 503 interrupted an affected trajectory segment. The complete eight-call segment was replayed under the frozen recovery procedure; all eight recovery calls returned valid structured responses, and recomputing the aggregate analysis produced the same primary estimate and confidence interval. The frozen primary result is unchanged.

H Scope of Information-Theoretic Interpretation

Exact likelihoods permit reference posterior and information calculations. These do not establish the LLM's latent beliefs, a causal relation between information gain and adaptation speed, or a tradeoff between evaluator quality and learnability. Such claims require separately specified interventions and, for oversight consequences, an external loss function.

I Cross-Provider and Model Robustness

Cross-model experiments use frozen subsets and prospectively chosen provider-native reasoning settings rather than tuning each provider to maximize the observed effect. The main text reports GPT-5.4-mini, Gemini 3.8 Flash, DeepSeek V4 Pro, and Qwen 3.8-27B REAL–NULL effects; Gemini additionally replicates the silent-switch and costly-information experiments.

DeepSeek V4 Pro. The audited DeepSeek run uses the official DeepSeek OpenAI-compatible endpoint, the frozen 64-pair subset spanning all 59 source pools, and low reasoning selected prospectively for operational cost before inspecting the run effect. All 512 REAL and 512 NULL records are complete and paired. The frozen analysis gives

$$\widehat{\Delta}_D = 0.1386, \qquad 95\%\,\mathrm{CI} = [0.0619, 0.2153].$$

Four terminal parser-invalid responses are retained as empty infeasible actions with FAIL and were not regenerated: three selected a historical claim identifier unavailable in the current packet and one returned truncated JSON. Separately, 110 saved responses followed provider-level transport/empty-response retries, representing 127 extra attempts before the saved scientific response. One exhausted NULL invocation was later resumed as a missing scheduled call. No unfavorable valid scientific action was selectively rerun.

Qwen 3.8-27B. The audited Qwen run uses OpenRouter with routing restricted to DeepInfra, model `qwen/qwen3.8-27b`, bfloat16 (BF16), medium reasoning, and the same frozen 64-pair subset. The run contains 512 REAL and 512 NULL calls across all 59 source pools. The final paired effect is

$$\widehat{\Delta}_D = 0.1652, \qquad 95\%\,\mathrm{CI} = [0.1106, 0.2198].$$

One terminal NULL parse failure was retained under the frozen invalid-response policy; the final provenance audit passed.

Model-family differences are descriptive; no causal architectural interpretation is made. We retain weaker or null completed replications rather than silently dropping them.

J PROMPTS AND MODEL/API SETTINGS

The controlled experiments use the system prompt `You are an ordered artifact selection agent.` The ordered-artifact schema is an object with required array field `ordered_claim_ids`, four entries drawn from the displayed claim-ID enum, and no additional properties. The provider schema does not set `uniqueItems`; the parser separately requires four distinct current-pool IDs and retains violations as invalid outputs. The active schema analogously requires one enumerated `action_id`. OpenAI, Gemini, and OpenRouter use strict JSON schema; DeepSeek uses JSON-object mode and appends a schema-derived valid JSON example. Terminal parser-invalid outputs are retained as empty infeasible actions with FAIL rather than scientifically rerun.

The actor-visible primary template discloses that the active evaluator is one of two candidate mechanisms, one biased toward earlier and one toward later positions, with equal prior probability and a fixed identity across eight rounds. It does not disclose the active mechanism, exact inspection weights, oracle actions, regret, posterior quantities, or future outcomes. At round t, it supplies the recipient's prior ordered claims (displayed ID, benchmark status, value) and PASS/FAIL outcomes, together with the current shuffled claim pool, threshold, two-inspection rule, and JSON-only instruction. NULL and YOKED use the same visible template; YOKED history contains the recipient's own actions and donor outcome bits only. Full rendered prompt templates and structured-output schemas are included in the supplementary artifact.

The switch template omits the stationarity sentence and says only that one of the earlier- or later-biased mechanisms is in use; it never mentions a switch or its timing. The costly-information template states the horizon, persistence condition, blinded current menu, and, at round 1 of a long episode, the blinded future menu. Later rounds expose only the actor's own action/outcome history. The feature-transfer prompt displays check IDs, feature pairs, and costs and requests the resolved-ID list; it does not display policy identity or the feature-weight rule.

Study/model	Provider	Reasoning	Temp.	Max	Output
Primary, switch, probe / GPT-5.4-mini-2026-03-17	OpenAI	medium	default	8192	strict schema
Primary, switch, probe / gemini-3.8-flash	Google	medium	default	8192	strict schema
Primary / deepseek-v4-pro	DeepSeek	low	default	8192	JSON object
Primary / qwen/qwen3.8-27b	OpenRouter (DeepInfra only)	medium	default	8192	strict schema
Transfer / gemini-3.8-flash	Google	medium	default	8192	strict schema

No run sets `top_p`; temperature is omitted and is provider-default. Configurations allow three provider attempts with 1, 2, and 4 second backoffs. Seeds are sent only to providers that support them; local prompt ordering and manifests remain deterministic. Runtime guards reject policy IDs and machine metadata in hidden prompts and reject switch-related language in the switch-study prompts. Prompt audits found no active identity, switch timing, oracle action, regret, posterior, or future outcome in actor-visible text.

K STATISTICAL INFERENCE AND SENSITIVITY ANALYSES

The primary analyses operate on the experimental units defined by the design rather than treating individual generations or rounds as independent observations. The primary REAL–NULL analysis preserves within-pair contrasts and accounts for shared source pools; The switch and costly-probe studies use their corresponding trajectory and microinstance structures.

For the primary REAL–NULL analysis, form paired round-level regret contrasts, fit saturated domain-by-policy-by-round cell means, and apply the frozen early-minus-late contrast with equal domain and policy weights. The covariance is the sum of source-pool and episode-pair cluster components minus their intersection, with finite-sample corrections. Each pool/episode intersection is a singleton in these schedules. Degrees of freedom are the minimum cluster count minus one; the GPT result uses 58. Neither GPT nor Gemini primary variance requires clipping. The secondary counterfactual contrasts use this same estimator after a sign transformation for late-minus-early utility changes. Their 95% intervals are unadjusted and descriptive; they do not replace the primary estimand.

This covariance models dependence by current source pool and episode. It does not establish absence of dependence through shared historical pools, semantic overlap between constructed pools, or provider-level effects. Claims of population-wide generalization should remain narrower than the controlled panel evidence. Exact-YOKED uses the same paired estimator, equal domain/policy weighting, two-way source-pool $\times$ episode-pair covariance, finite-sample corrections, and t_{58} interval on its prospectively selected 160 pairs. As an offline dependence sensitivity, Gaussian wild multipliers combine an episode shock with equal-weight shocks for all eight source pools in its history. Across modeled source-history variance shares from 0.25 to 1.0, the resulting intervals range from $[0.1515, 0.2455]$ to $[0.1563, 0.2414]$, compared with the frozen primary interval $[0.1429, 0.2545]$; this modeled sensitivity does not replace the pre-specified covariance. The silent-switch unit is a ten-round trajectory; its contrast is round-6 new-policy regret minus mean round-9/10 new-policy regret, equally weighted over domain and transition direction, with the same two-way cluster-robust covariance over 59 pools and 64 trajectories and a t_{58} interval. The costly-probe unit is the microinstance. Its difference-in-differences is recomputed under 2,000 seed-47 bootstrap resamples of all 24 microinstances, and the 2.5th/97.5th percentiles form the interval. Sequential follow-through summaries are descriptive. The feature-transfer experiment follows its own pre-specified analysis plan rather than the primary REAL–NULL analysis. It uses the equal-cell marginal mean from a saturated six-stratum regression with HC3 heteroskedasticity-robust covariance; with eight matched units per cell this leaves $48 - 6 = 42$ residual degrees of freedom. The final analysis gives $\widehat{\Delta}_R = 0.1663$ with $\mathrm{SE} = 0.0336$ and 95% CI $[0.0985, 0.2342]$. Evaluator specificity and the generic/specific decomposition remain secondary. Effect sizes and confidence intervals are the primary summaries; exploratory analyses do not replace the pre-specified estimands.

L Reproducibility, Provenance, and Qualitative Trajectories

Every paper number maps to an immutable raw record, frozen configuration, analysis entry point, and figure-data row. Completeness audits reject duplicate or missing pair–round keys and recompute utilities and regret from source pools. The primary GPT panel contains 7,552 records with no parser failures (44 REAL and 69 NULL extra provider attempts); Gemini contains 1,024 with no parser failures or retries. DeepSeek contains 1,024 records, four retained terminal invalid outputs, and 127 extra attempts underlying 110 saved responses; Qwen contains 1,024 and one retained terminal invalid output. GPT exact-YOKED contains 1,280 records, no parser failures, and eight transport retries. Each GPT/Gemini switch run contains 640 complete records with no parser failure (14 and zero extra attempts), and each costly-information run contains 960 complete records with no parser failure (zero and three extra attempts). Structural transfer contains 864 hidden calls; one HTTP 503 caused the frozen eight-call segment replay described above.

The analysis environment is Python 3.10 with NumPy $\geq$ 1.23, pandas $\geq$ 2.0, SciPy $\geq$ 1.10, Matplotlib $\geq$ 3.7, PyYAML $\geq$ 6.0, Pydantic $\geq$ 2.0, PyArrow $\geq$ 14.0, OpenAI SDK $\geq$ 1.40, pytest $\geq$ 8.0, and tabulate $\geq$ 0.9. Offline entry points are `analyze_p0_existing.py` for primary/specificity/references, `audit_p0_remaining.py` for switch/probing, `analyze_yoked_robustness.py` for yoking, `consolidate_final_results.py` for the canonical ledger, and `render_final_paper_figures.py` for Figures 2–4. Qualitative records are selected by deterministic identifiers rather than visual appeal.

Saved total-token counts are 722,498 (transparent capability), 34,490,989 (GPT primary REAL/NULL), 3,717,523 (Gemini primary replication), 6,414,949 (DeepSeek), 5,581,317 (Qwen), 5,369,810 (GPT YOKED), 2,893,318 and 2,323,093 (GPT/Gemini switch), 2,628,168 and 2,621,287 (GPT/Gemini costly information), and 7,331,508 (hidden structural transfer). The capability-only structural-transfer artifact does not expose a complete token aggregate and is therefore not assigned one here.