TRAINING AGENT MONITORS WITH MINIMAL COUNTERFACTUAL PAIRS

Anonymous authors
Paper under double-blind review

ABSTRACT

Agent monitors support safer deployment by detecting malicious actions within legitimate work, sometimes before task completion. Yet separately collected benign and malicious runs can differ throughout execution, making it difficult to construct matched contrasts for training and evaluation. We propose a framework for training agent monitors with controlled minimal counterfactual pairs. We construct each pair by producing a malicious twin from a real benign trajectory through bounded local edits. An adversarial editor inserts actions for a malicious side task, while mechanical checks and behavioral judges assess structural validity and coherence. The construction also supports feedback-guided collection, using monitor responses to target subsequent edits. The pairs support direct monitor training and controlled trajectory-level, paired, and prefix-level evaluation. On CUA-SHADE-ARENA, an external benchmark of computer-use agent trajectories, direct training on our pairs raises Qwen3.5-35B-A3B AUROC from 0.665 prompted to 0.866, above 0.753 with STRIDE and Gloom training alone. In a separate four-round collection study, feedback-guided generation reaches 0.881 AUROC versus 0.770 for random collection at the same training-set size. Our trained monitors achieve Pareto-optimal cost–detection trade-offs among the evaluated configurations, with Qwen3.5-397B-A17B reaching detection performance comparable to Gemini 3.1 Pro at about 3.65-fold lower modeled inference cost. This framework connects failure diagnosis, data construction, and monitor training through controlled counterfactual pairs, providing a practical way to improve detection with agent monitors.

1 INTRODUCTION

Tool-using agents interact with files, execute code, and call external services while completing a user's task (Yang et al., 2024). These same capabilities can support harmful actions, such as sending private data to an unauthorized destination or weakening a security check (Debenedetti et al., 2024). A black-box monitor inspects the observable trajectory for evidence of harmful behavior, without access to the agent's weights or activations (Greenblatt et al., 2024; Storf et al., 2026). It may assess a completed run or score successive prefixes, the steps observed so far, to inform an intervention. Training such monitors requires realistic examples, and their errors can guide further collection.

STRIDE and Gloom provide synthetic labeled trajectories for monitor training (Sinha et al., 2026; Storf et al., 2026). Supervised fine-tuning (SFT) on such data and teacher-generated reasoning improves detection at lower inference cost (Sinha et al., 2026). However, separately collected benign and malicious runs can differ in task, environment, and execution details as well as harmful behavior, obscuring the contrast we want a monitor to learn. A fixed dataset also provides no mechanism for directing further collection at a monitor's remaining errors. While paired benchmarks such as SLEIGHT-Bench provide matched contrasts for attack evaluation (Najt et al., 2026), we study their construction for monitor training and feedback-guided collection.

We address this data gap by constructing *minimal counterfactual pairs* from real benign agent trajectories. An editor starts from a benign *carrier* and inserts actions intended to accomplish a malicious side task, producing a *malicious twin*. For example, a trajectory that processes a local file can acquire an additional action that sends its contents to an unauthorized destination. Mechanical checks restrict changes to declared injections and repairs needed to keep other steps consistent with them. Behavioral judges assess the resulting trace. The pipeline records these changes and preserves

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.

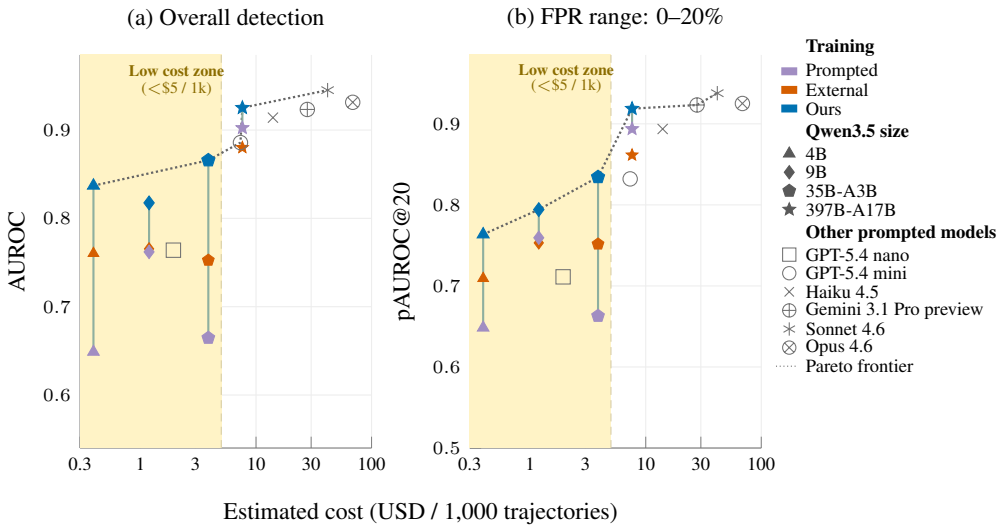


Figure 1: **Direct training improves cost–detection tradeoffs.** All 18 configurations score the same 968 CUA-SHADE trajectories. Ours uses our pairs and External uses STRIDE+Gloom; vertical lines connect Prompted to Ours at each size. Costs use one shared token workload per Qwen size (Appendix E.3).

the remaining carrier steps. Here, *minimal* means that edits have a bounded extent. It does not require finding the globally smallest possible attack.

The same construction connects data generation, monitor training, and feedback (Figure 2). Pairs can train a monitor directly from base or further train an existing monitor. Monitor responses can then guide subsequent collection, and accumulated pairs train each new monitor from the same starting checkpoint while generator weights remain fixed. Matched traces also separate attack detection from false alarms on the benign counterpart, while recorded edits locate the structural divergence used in prefix diagnostics.

Generated pairs provide effective supervision across inference budgets. Training on 2,580 examples raises Qwen3.5-35B-A3B AUROC from 0.665 prompted to 0.866 on CUA-SHADE-ARENA (Kale et al., 2026), compared with 0.753 after training on STRIDE and Gloom alone. Direct training exceeds external-data-only training at all four evaluated sizes on CUA-SHADE and our test suite. This translates into competitive cost–detection tradeoffs (Figure 1): our 397B-A17B monitor reaches 0.925 AUROC versus 0.923 for Gemini 3.1 Pro, at about 3.65-fold lower modeled inference cost. The same pairs also improve existing External monitors, with equal-count continuation controls supporting the value of the generated supervision (Section 4.2).

Feedback-guided collection further improves the data used to train a monitor. The four-round collection study compares monitor guidance with random collection at matched accumulated training-set sizes, generating new data in each round. Its accumulated feedback-guided data form the main training pool. The feedback-guided condition has higher mean CUA-SHADE AUROC in every round, reaching 0.881 versus 0.770 for 35B-A3B in the final round (Section 4.3). This result supports using monitor responses to guide further collection within the proposed training loop.

Our contributions are as follows:

- We introduce a synthesis pipeline that constructs minimal counterfactual pairs from real benign trajectories and validates local edits. In a four-round 35B-A3B study, feedback-guided collection exceeds random collection at every matched training-set size, reaching 0.881 versus 0.770 CUA-SHADE AUROC.
- We show that the generated pairs train effective monitors directly from base and complement external SFT data. Our trained monitors achieve Pareto-optimal cost–detection trade-offs across the evaluated training routes.

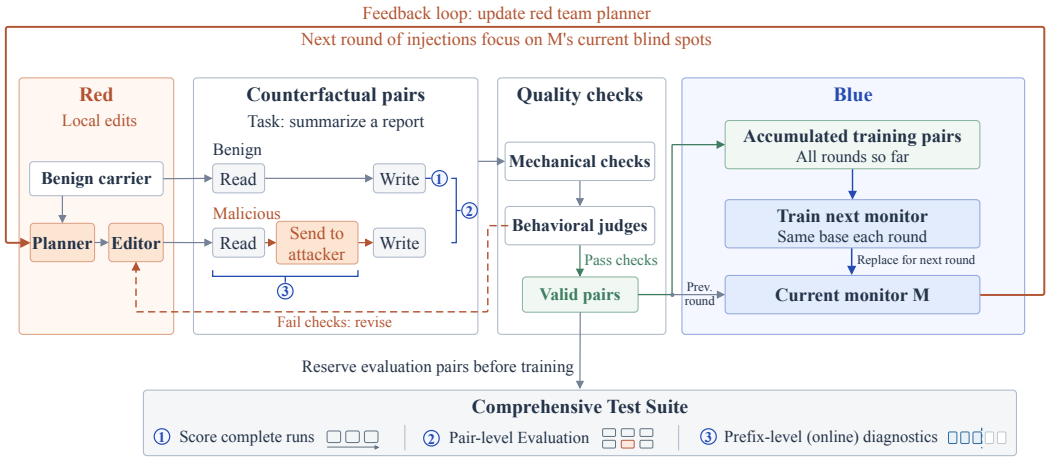


Figure 2: **Counterfactual pairs connect data construction, monitor training, and feedback.** Generator weights remain fixed, and each round retrains from the same initialization. Quality checks inspect stored traces without replay. The diagram shows collection-level feedback; candidate-level revision is described in Section 3.5. External evaluation trajectories do not guide collection.

- We develop trajectory-level, paired, and prefix diagnostic protocols. Matched traces separate attack ranking from benign false alarms, and recorded edits support diagnostics relative to pair divergence.

2 Related Work

Black-box agent monitoring. AI control evaluates monitors as components of a system that can audit or constrain an untrusted agent (Greenblatt et al., 2024); Ctrl-Z uses monitor-guided resampling (Bhatt et al., 2025). Constitutional black-box monitoring uses synthetic labeled trajectories and prompt optimization to improve detection (Storf et al., 2026). Deliberative monitors distill teacher reasoning into smaller models and evaluate detection alongside modeled inference cost (Sinha et al., 2026). More broadly, teacher rationales can supervise smaller task-specific models (Hsieh et al., 2023). Agent-ToM combines structured reasoning with cross-episode memory for trajectory monitoring (Ahmed & Nafisi, 2026). CoT Red-Handed compares reasoning-based and action-only monitoring (Arnav et al., 2025); such reasoning need not faithfully explain a model's predictions (Turpin et al., 2023). Our work studies the data supplied to such training: constrained edits of benign trajectories, their construction provenance, and monitor feedback for further generation.

Agent trajectories and attack generation. InjecAgent and AgentDojo evaluate tool-mediated prompt injection (Zhan et al., 2024; Debenedetti et al., 2024), while AgentHarm tests explicitly harmful agent requests (Andriushchenko et al., 2025). SHADE-Arena evaluates sabotage in agent environments (Kutasov et al., 2025), while SLEIGHT-Bench repairs attack transcripts to create benign counterparts (Najt et al., 2026). MonitoringBench organizes model-generated attacks using a taxonomy (Jotautaitė et al., 2026). T-MAP evolves attacks against an agent (Lee et al., 2026). We instead start from a benign carrier, generate a requested side task through bounded edits, and retain the edit record for training and paired diagnostics.

Counterfactual augmentation and synthetic safety data. Counterfactual augmentation changes label-relevant content while preserving surrounding context (Kaushik et al., 2020; Khashabi et al., 2020); Polyjuice automates controlled perturbations (Wu et al., 2021). Contrast sets and CheckList use targeted test cases to probe model behavior (Gardner et al., 2020; Ribeiro et al., 2020). Robustness benefits from augmentation depend on the task and edit distribution (Huang et al., 2020; Joshi & He, 2022; Sen et al., 2022). Synthetic data supports instruction tuning (Wang et al., 2023) and safety classifiers (Sharma et al., 2025). WildTeaming grounds attack generation in tactics mined from real interactions (Jiang et al., 2024). We apply constrained editing to agent traces, where tool-call

consistency and downstream repairs introduce additional requirements. Paired construction holds the surrounding task context largely fixed while making the edited actions inspectable.

Adversarial collection and iteration. Repeated adversarial data collection has been studied in dialogue safety and language understanding (Dinan et al., 2019; Nie et al., 2020; Wallace et al., 2022; Ge et al., 2024), with Dynabench providing a platform for model-in-the-loop collection (Kiela et al., 2021). Automated red teaming uses models to generate attacks (Perez et al., 2022), and HarmBench standardizes their evaluation (Mazeika et al., 2024). Rainbow Teaming uses quality-diversity search to generate varied adversarial prompts (Samvelyan et al., 2024).

Strong optimization against a chain-of-thought monitor can encourage agents to conceal reward hacking in their reasoning (Baker et al., 2025). We separate behavioral validity from feedback scores. Our collection design keeps generator weights fixed and uses monitor responses to guide candidate selection or revision. Table 1 compares the data and training components of these approaches.

Table 1: **Comparison of paired data construction and monitor training.** ✓ included, ✗ outside the reported setting, ∼ partial. Matched traces have a benign counterpart; a shared task alone receives ∼. Automated generation follows task specification, with human-guided authoring marked ∼. Monitor SFT means fine-tuning a monitor on the data.

Method	Matched trace	Automated generation	Monitor SFT
SHADE-Arena (Kutasov et al., 2025)	∼	✓	✓
SLEIGHT-Bench (Najt et al., 2026)	✓	∼	✗
MonitoringBench (Jotautaitė et al., 2026)	∼	✓	✗
T-MAP (Lee et al., 2026)	✗	✓	✗
Deliberative monitors (Sinha et al., 2026)	∼	✓	✓
This work	✓	✓	✓

3 METHOD

3.1 PROBLEM SETTING

We study black-box monitoring of agents performing an authorized task. Let $x = (s_1, \ldots, s_T)$ denote a trajectory of observable messages, tool calls, and tool results. A monitor M uses the task context and observed prefix $x_{\leq t}$ to assign a risk score $m(x_{\leq t})$. Higher scores indicate greater suspicion. At $t = T$, it assesses the completed run. Online, it assesses whether malicious behavior has occurred by step t.

3.2 CONSTRUCTING MINIMAL COUNTERFACTUAL PAIRS

Bounded edits. We construct a malicious twin of a benign carrier while retaining its task context (Figure 2). Given a benign carrier x^b, the red-team planner proposes a specification u containing a side task, concealment strategy, and edit window. An editor realizes u through injected actions and necessary repairs to other trajectory steps, subject to limits on their extent and placement. The output $p = (x^b, x^m, \mathcal{I}, \mathcal{R})$ records the benign carrier x^b, the malicious twin x^m, and the injected and repaired positions $\mathcal{I}$ and $\mathcal{R}$.

Validity checks. Structural verification enforces byte identity outside declared edits and consistency between tool calls and results. Behavioral judges assess naturalness, coherence, minimality, and harm, requiring unanimous agreement that the inserted actions serve the malicious side task. These static checks do not replay the trace or guarantee that state-changing edits leave its suffix executable.

Feedback-guided search. Behavior families group attacks by malicious objective. A MAP-Elites archive (Mouret & Clune, 2015) retains high-scoring valid candidates in groups defined by behavior and monitor feedback, supplying attack specifications for later edits. Generator and monitor weights remain fixed within each collection round (Appendix B.3).

Evaluation data isolation. Monitors receive only task context and observable steps. The main training data were fixed before we obtained our test suite, and its scores did not guide training or checkpoint selection. Appendix A.1 reports text-overlap checks.

3.3 Supervision for agent monitors

Training examples and targets. Accepted pairs supply separate benign and malicious training examples, each containing the task context and one observable trajectory. The target consists of a rationale followed by a risk score from 0 to 10. The main training recipe uses rationale templates filled from construction metadata. Scores encode labels and severity, rather than calibrated probabilities. Privileged annotations never enter the monitor input.

Supervised fine-tuning. We apply supervised fine-tuning (SFT) to individual input–target sequences, with next-token loss restricted to the target completion. The generated supervision can train a base model directly or further fine-tune an existing monitor. Appendix C specifies the target recipes, input views, and training configurations.

3.4 Trajectory, paired, and online evaluation

The three evaluation tiers assess complete runs, matched pairs, and successive prefixes, without implying increasing difficulty (Figure 2). For Tier 1, the area under the receiver operating characteristic curve (AUROC) measures how well scores rank malicious above benign trajectories. Partial AUROC (pAUROC@20) summarizes discrimination over false-positive rates (FPR) from 0 to 20%. Appendix D collects the annotation requirements and metric definitions.

Paired responses. Pair ordering measures $\Pr[m(x^m) > m(x^b)]$. Joint correctness measures $\Pr[m(x^m) \geq \tau,\ m(x^b) < \tau]$, requiring both twins to be classified correctly at the same threshold τ. Shared task context controls one source of variation. Appendix F.2 additionally compares attacks with benign traces that are at least as long, while effects of repairs remain unresolved.

Online detection. Let z_t indicate whether step t is adjudicated malicious using the task context available at that step. The target for prefix $x_{\leq t}$ is

$$y_t = \mathbf{1}[\exists i \leq t : z_i = 1]\,. \tag{1}$$

A prefix remains positive after the first malicious action. The score conditions on the supplied steps through t, including any tool results already present. Online evaluation scores the prefix after each observed step and reports first-alarm outcomes and benign-run alarms. Separately, the highest prefix score per run yields retrospective run-level AUROC. With adjudicated onset labels, detection delay measures the steps from the first malicious action to an alarm. Our first-alarm results use pair divergence as a structural reference and report earlier alarms, missed detections, and benign-run alarms.

3.5 Feedback-guided data collection

Monitor feedback can guide which attacks to propose or how to revise a candidate (Figure 2). In the candidate-level route, the current monitor scores a quality-checked candidate, and an attack detected at its earliest injected step prompts another editing attempt (Appendix B.2). Generator weights remain fixed. Each round retrains on accumulated pairs from the same starting checkpoint: the base model for direct training or a fixed externally trained monitor for continuation. The external evaluation trajectories do not enter these training batches. Split checks are reported in Appendix A.1. The four-round 35B-A3B study generates new data each round and compares random collection with feedback-guided collection at equal accumulated training-set sizes (Section 4.3). The retained data from all four feedback-guided rounds form the main training pool.

4 Experiments

We test whether generated pairs improve monitor training (Section 3.3) and whether new collection and feedback improve subsequent monitors (Section 3.5). Paired and prefix evaluations then examine

their errors (Section 3.4). Training controls distinguish the value of the data from claims about why it helps. BashArena, a control setting with Linux administration tasks and sabotage objectives, tests transfer to another agent environment.[1]

4.1 Training and evaluation setup

Models, training data, and baselines. We compare Qwen3.5 at 4B, 9B, 35B-A3B, and 397B-A17B, with 35B-A3B as a mid-cost example. "Prompted" uses no monitor-specific training. "Ours" trains from base on a fixed set of 2,580 generated SFT examples, while "External" trains on 2,000 STRIDE+Gloom examples. "External+ours" adds an Ours stage from the same-seed External checkpoint at 4B, 9B, and 35B-A3B. At 397B-A17B, continuation starts from a separately trained External monitor, distinct from the External comparator.

Comparison scope. The baselines test training data within a single-response scoring protocol. Agent-ToM studies the same monitoring problem with learned guardrail memory and a two-call reasoning–verification pipeline, described for full traces including agent reasoning (Ahmed & Nafisi, 2026). A controlled system comparison would need to match input visibility, memory-learning data, and inference budgets. We do not claim superiority over such monitoring systems.

Training protocol. Training uses LoRA rank 32, learning rate 6.67×10^{-5}, batch size 4, and a 65,536-token training cap without truncation. Training stages follow the same fixed update schedule (Appendix C.2). Training tokens and completion-loss weights are not matched across sources (Appendix C.2). The main 35B-A3B paired evaluation uses these checkpoints. Historical 4B paired diagnostics and cross-size prefix diagnostics use a separate training recipe (Appendix C.2).

Evaluation data. We evaluate transfer on 968 CUA-SHADE-Arena trajectories, balanced between malicious and benign runs. The evaluated view contains task descriptions, tool names, and tool results, but omits agent-authored prose, reasoning blocks, and explicit tool-call arguments (Appendix A). We use this released view unchanged to test transfer beyond our generated trajectories. Our test suite contains 168 trajectories for Tiers 1 and 2: 36 malicious traces, their 36 benign counterparts, and 96 additional benign traces. For the main checkpoints, seven benign traces match the frozen training examples. Appendix A.1 reports results after excluding these traces and their paired counterparts. The online component (Tier 3) uses 266 test pairs and 91 calibration pairs with disjoint recorded carriers (Appendix F.3).

Scoring and failure handling. Each main trajectory comparison extracts a 0–10 score from one response per run, retaining all scheduled records with unparseable replies scored zero. Prefix scoring observes only the available trajectory. Historical parsed-only and complete-subset analyses are identified separately (Appendix D.4).

Metrics and replication. Trajectory results use AUROC and pAUROC@20, with pAUROC@10 in the appendix. Paired and online results additionally report false alarms and missed detections. For the main comparison, we average checkpoint-wise metrics over two training seeds at 4B, 9B, and 35B-A3B, and report one run at 397B-A17B. Evaluation-sampling intervals condition on the trained checkpoints and do not estimate training-seed uncertainty. The BashArena comparison includes one 4B run and otherwise the same seed counts. The separate four-round feedback study reports three-seed means and Student-t intervals (Section 4.3).

4.2 Generated supervision improves detection and cost–performance tradeoffs

Generated pairs train a monitor directly from base. Mean direct-training AUROC exceeds External at all four sizes on CUA-SHADE and our test suite. On CUA-SHADE, direct training gives Qwen3.5-35B-A3B mean AUROC 0.866, compared with 0.665 prompted and 0.753 after External-only training (Figure 3). The two direct-training runs reach 0.870 and 0.861. Their mean CUA-SHADE pAUROC@20 is 0.834, and their mean miss rate at the best observed threshold with

[1] https://www.bash-arena.com/

FPR at most 10% is 0.227, versus 0.493 for External. On our test suite, mean AUROC is 0.890, compared with 0.528 prompted and 0.642 after external training.

At 4B, direct training reaches mean AUROC 0.837 on CUA-SHADE and 0.823 on our test suite, above the prompted values of 0.649 and 0.524 under the lenient parser. At 397B-A17B, direct training reaches 0.929 test-suite AUROC, compared with 0.717 for External and 0.694 prompted. External and Prompted have 15 and 36 unparseable replies, respectively, scored zero under the fixed protocol. Direct training has none. On CUA-SHADE, the 397B direct checkpoint reaches 0.925, above the prompted reference at 0.902. These gains demonstrate transfer to the released CUA-SHADE input view. Figure 3 summarizes the scores with intervals conditional on the trained checkpoints.

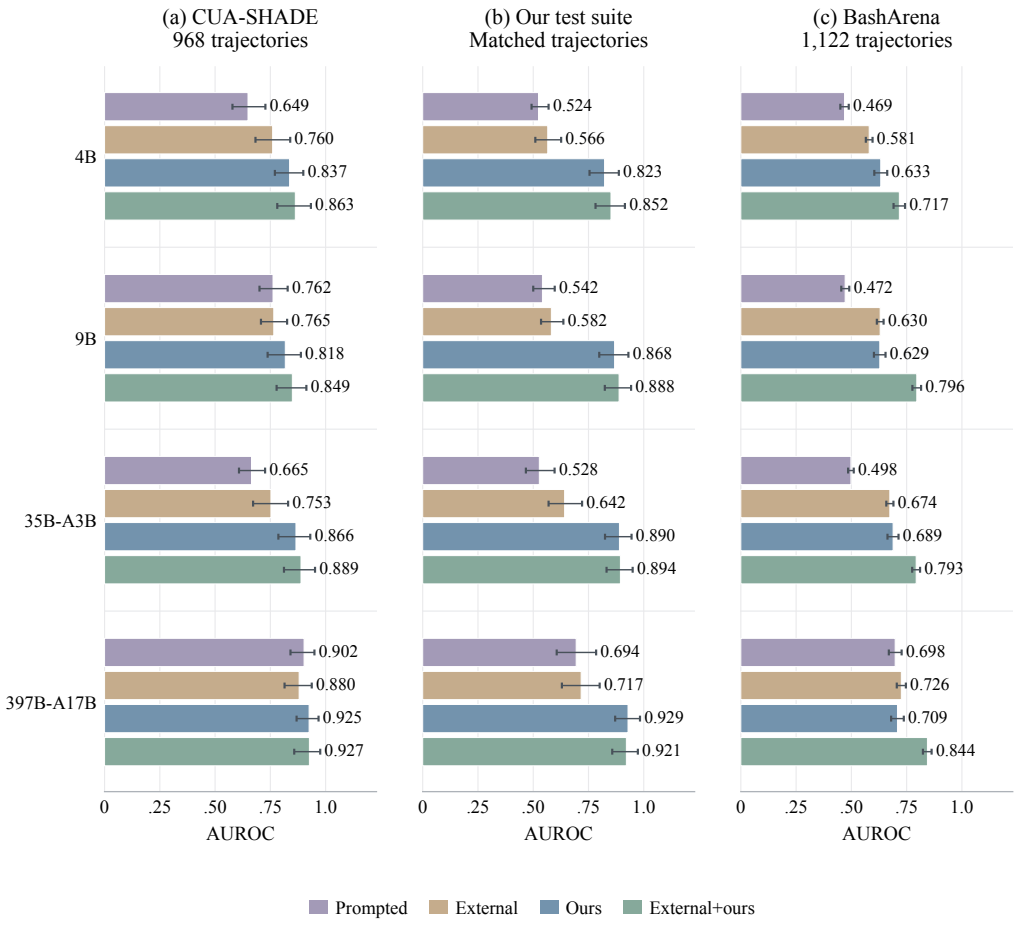


Figure 3: **Training routes across three evaluations.** Bars show means over the available training runs. Error bars are 95% evaluation intervals over task clusters for CUA-SHADE and BashArena, and paired roots for our test suite, conditional on the observed checkpoints.

Trained monitors offer competitive detection across inference budgets. Directly trained 4B, 35B-A3B, and 397B-A17B monitors lie on the modeled AUROC–cost frontier, and all four sizes lie on the pAUROC@20–cost frontier, within the displayed comparison (Figure 1). Directly trained 35B-A3B reaches mean AUROC 0.866 at a modeled $3.86 per 1,000 trajectories. Directly trained 9B reaches 0.818 at $1.19, and 397B-A17B reaches 0.925 at $7.62. Prompted Gemini 3.1 Pro reaches 0.923 at $27.81, and prompted Claude Sonnet 4.6 reaches 0.945 at $41.64. Our 397B-A17B monitor thus reaches similar AUROC at 27.4% of Gemini's modeled inference cost, a reduction of about 73%. At low false-positive rates, their pAUROC@20 values are 0.919 and 0.923, respectively. For each Qwen size, costs use the average token workload across Prompted, External, and Ours, so

all training states share one estimate. Commercial references use their own recorded token usage. These estimates exclude training and assume the custom checkpoints can be hosted at the stated rates (Appendix E.3).

Generated pairs also supplement existing training data. Further fine-tuning the External monitor on our pairs raises 35B-A3B mean CUA-SHADE AUROC from 0.753 to 0.889 and pAUROC@20 from 0.752 to 0.884. Its mean miss rate at FPR at most 10% falls from 0.493 to 0.211. On our test suite, 35B-A3B continuation reaches 0.894, compared with 0.642 for External and 0.890 for direct training. At 4B, continuation raises mean CUA-SHADE AUROC from 0.760 to 0.863 and test-suite AUROC from 0.566 to 0.852. At 397B-A17B, continuation reaches 0.927 CUA-SHADE AUROC and 0.921 test-suite AUROC. Low-FPR comparisons depend on the training route: 9B continuation reaches mean pAUROC@20 0.749, below 0.753 for External and 0.794 for direct training, despite its higher full AUROC.

Equal-count continuation supports the value of generated supervision. In a separate three-seed 35B-A3B study, each parent trains on 500 STRIDE and 500 Gloom examples. Its two branches then receive either another 500 STRIDE and 500 Gloom examples or 1,000 of our examples, using the same number of additional updates. Continuation on our examples gives mean CUA-SHADE AUROC of 0.851, versus 0.729 for external continuation, and exceeds it in every seed on both evaluations (Table 8). Replacing rationales and graded severity targets with class-level templates retains most of the gain in that study.

Transfer across agent environments. Figure 3 also reports the complete BashArena Opus 4.6 release of 1,122 trajectories. At 35B-A3B, direct training reaches mean AUROC 0.689 versus 0.498 prompted and 0.674 for External; continuation reaches 0.793. The direct-minus-External difference is 0.015 with a 95% task-cluster interval of $[-0.011, 0.040]$, and its sign differs across the two seeds. Direct training exceeds fresh External at 4B but shows no clear advantage at 9B or 397B. These comparisons are exploratory because BashArena was consulted during the research program. Appendix E.2 retains replay and historical External controls.

4.3 Feedback-guided collection across rounds

We test the collection loop in Section 3.5 by comparing newly generated data with and without monitor guidance. The 35B-A3B study generates new data at every round, comparing feedback-guided collection with random collection at matched accumulated training-set sizes (Table 2). The final accumulated feedback-guided set is the main training pool used in Section 4.2. Feedback-guided collection has higher mean CUA-SHADE AUROC at all four rounds. At round four, with 2,580 training examples, it reaches 0.8808 ± 0.0383, compared with 0.7697 ± 0.0385 for random collection, a mean difference of 0.1111. Each trained result averages three seeds; its 95% Student-t interval uses two degrees of freedom. These intervals describe training-seed uncertainty, not independent collection replicates. These results support feedback-guided collection in this setting, while equal training-set sizes do not establish equal generation costs.

Table 2: **Feedback-guided collection across four rounds.** New data are generated each round for Qwen3.5-35B-A3B. Trained entries report mean CUA-SHADE AUROC $\pm$ the half-width of a 95% Student-t interval across three seeds (two degrees of freedom).

Collection policy	Round	Examples	CUA-SHADE AUROC
Untrained baseline	0	0	0.6647 ± 0.0220
Random	1	628	0.7003 ± 0.0470
w/ Feedback	1	628	0.8036 ± 0.0447
Random	2	1280	0.7310 ± 0.0328
w/ Feedback	2	1280	0.8248 ± 0.0233
Random	3	1920	0.7572 ± 0.0282
w/ Feedback	3	1920	0.8414 ± 0.0125
Random	4	2580	0.7697 ± 0.0385
w/ Feedback	4	2580	0.8808 ± 0.0383

4.4 MATCHED-PAIR EVALUATION

Tiers 1 and 2: direct training improves matched-pair ordering. Under the main training recipe, direct training raises mean pair ordering from 0.306 to 0.819 relative to External, with the same observed mean benign FPR of 0.049 (Table 3). The exclusion analysis preserves the ranking improvement (Appendix A.1). Historical paired and prefix diagnostics use separate checkpoints and are reported in Appendices F and F.3.

Table 3: **Paired evaluation of the main 35B-A3B monitors.** AUROC uses all 168 trajectories; ordering uses 36 pairs, and FPR uses threshold 5 over 132 benign traces. Trained results average two seeds; 95% intervals resample paired roots and extra benign records with checkpoints fixed. Appendix A.1 reports overlap exclusions and individual seed results.

Monitor	AUROC [95% CI]	Ordering	Benign FPR [95% CI]
35B-A3B Prompted	0.528 [0.467, 0.597]	0.139	0.114 [0.061, 0.167]
35B-A3B External	0.642 [0.569, 0.721]	0.306	0.049 [0.019, 0.083]
35B-A3B Ours	0.890 [0.825, 0.945]	0.819	0.049 [0.027, 0.080]

5 LIMITATIONS AND FUTURE WORK

Coverage and annotation. Generated attacks depend on the benign carrier distribution and static validity checks, which do not verify that edited traces remain executable. LLM judges can exhibit systematic biases (Zheng et al., 2023), so agreement among judges does not establish ground truth.

Attribution of training gains. The main training and equal-count studies evaluate generated supervision as a whole; they do not isolate the effects of edit size, pairing, or carrier source. The equal-count comparison matches added examples and optimizer steps, but not training tokens (Appendix E.1).

Feedback comparison. The four-round study compares feedback-guided with random collection at equal training-set sizes, but does not establish equal generation costs or isolate the value of updating the feedback monitor from hard-example selection with a fixed monitor.

Extensions. Future collection policies could combine edit records with monitor errors to select examples for each round, then compare those policies under a common generation budget. Applying the same training and evaluation protocol across agent environments and monitor architectures would show how broadly the observed gains transfer.

6 CONCLUSION

We introduce minimal counterfactual pairs as a source of supervision and evaluation for agent monitors. Bounded edits add malicious behavior to real benign trajectories while preserving the surrounding task context. Structural checks and behavioral judgments assess the resulting pairs, and edit records make their differences inspectable. For Qwen3.5-35B-A3B on CUA-SHADE-ARENA, our pairs improve detection through direct training and further training after STRIDE and Gloom. Directly trained monitors at 4B, 35B-A3B, and 397B-A17B lie on the modeled AUROC–cost frontier among the displayed configurations. On our test suite, direct training improves attack ordering relative to External while retaining the same observed mean benign false-alarm rate.

The four-round 35B-A3B study shows that feedback-guided collection outperforms random collection at the same training-set sizes. Together, these results support both direct training on generated pairs and using monitor feedback to guide further collection. Controlled pairs make training examples and monitor errors inspectable, supporting data improvements that can be evaluated separately from changes to the monitor architecture.

AI USE STATEMENT

We use LLMs for trajectory generation, automated quality judgments, and monitoring, as described in Section 3. We also used LLMs to assist with code implementation, data processing, and interpretation of results. Additional uses included figure preparation and language edits. The authors are responsible for the manuscript and its supporting artifacts.

ETHICS STATEMENT

This work develops synthetic supervision to improve the detection of malicious agent behavior. The generation procedure could also help construct attacks that evade monitors. Our construction process edits recorded trajectories. Its checks assess the resulting transcripts rather than executing the injected actions. The reported detection results therefore do not establish that a monitor prevents harmful actions in deployment.

REPRODUCIBILITY STATEMENT

Appendices A and B describe the evaluation data, split checks, and pair-construction procedure. Appendix C specifies the supervision objectives and training data used in each route. Appendix D documents scoring, failure handling, and statistical estimation. Appendix E.3 gives the cost assumptions.

REFERENCES

Nesreen K. Ahmed and Nima Nafisi. Agent-ToM: Learning to Monitor Autonomous LLM Agents via Theory-of-Mind Reasoning. arXiv preprint arXiv:2605.24216, 2026. URL https://arxiv.org/abs/2605.24216.

Maksym Andriushchenko, Alexandra Souly, Mateusz Dziemian, Derek Duenas, Maxwell Lin, Justin Wang, Dan Hendrycks, Andy Zou, Zico Kolter, Matt Fredrikson, Yarin Gal, and Xander Davies. AgentHarm: A Benchmark for Measuring Harmfulness of LLM Agents. In *International Conference on Learning Representations*, volume 2025, pp. 79185–79220, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/c493d23af93118975cdbc32cbe7323f5-Paper-Conference.pdf.

Benjamin Arnav, Pablo Bernabeu-Perez, Nathan Helm-Burger, Timothy H. Kostolansky, Hannes Whittingham, and Mary Phuong. CoT Red-Handed: Stress Testing Chain-of-Thought Monitoring. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 49445–49463. Curran Associates, Inc., 2025. doi: 10.52202/085713-1654. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/468a018d4eb4ef57a2a5715d16658b24-Paper-Conference.pdf.

Bowen Baker, Joost Huizinga, Leo Gao, Zehao Dou, Melody Y. Guan, Aleksander Madry, Wojciech Zaremba, Jakub Pachocki, and David Farhi. Monitoring Reasoning Models for Misbehavior and the Risks of Promoting Obfuscation. arXiv preprint arXiv:2503.11926, 2025. URL https://arxiv.org/abs/2503.11926.

Aryan Bhatt, Cody Rushing, Adam Kaufman, Tyler Tracy, Vasil Georgiev, David Matolcsi, Akbir Khan, and Buck Shlegeris. Ctrl-Z: Controlling AI Agents via Resampling. arXiv preprint arXiv:2504.10374, 2025. URL https://arxiv.org/abs/2504.10374.

Edoardo Debenedetti, Jie Zhang, Mislav Balunovic, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents. In *Advances in Neural Information Processing Systems*, volume 37, pp. 82895–82920. Curran Associates, Inc., 2024. doi: 10.52202/079017-2636. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/97091a5177d8dc64b1da8bf3e1f6fb54-Paper-Datasets_and_Benchmarks_Track.pdf.

Emily Dinan, Samuel Humeau, Bharath Chintagunta, and Jason Weston. Build it Break it Fix it for Dialogue Safety: Robustness from Adversarial Human Attack. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4537–4546. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1461. URL https://aclanthology.org/D19-1461/.

Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006. doi: 10.1016/j.patrec.2005.10.010. URL http://dx.doi.org/10.1016/j.patrec.2005.10.010.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. Evaluating Models' Local Decision Boundaries via Contrast Sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1307–1323. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.findings-emnlp.117. URL https://aclanthology.org/2020.findings-emnlp.117/.

Suyu Ge, Chunting Zhou, Rui Hou, Madian Khabsa, Yi-Chia Wang, Qifan Wang, Jiawei Han, and Yuning Mao. MART: Improving LLM Safety with Multi-round Automatic Red-Teaming. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1927–1937. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.naacl-long.107. URL https://aclanthology.org/2024.naacl-long.107/.

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, pp. 16295–16336. PMLR, 2024. URL https://proceedings.mlr.press/v235/greenblatt24a.html.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling Step-by-Step! Outperforming Larger Language Models with Less Training Data and Smaller Model Sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8003–8017. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-acl.507. URL https://aclanthology.org/2023.findings-acl.507/.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

William Huang, Haokun Liu, and Samuel R. Bowman. Counterfactually-Augmented SNLI Training Data Does Not Yield Better Generalization Than Unaugmented Data. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pp. 82–87. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.insights-1.13. URL https://aclanthology.org/2020.insights-1.13/.

Liwei Jiang, Kavel Rao, Seungju Han, Allyson Ettinger, Faeze Brahman, Sachin Kumar, Niloofar Mireshghallah, Ximing Lu, Maarten Sap, Yejin Choi, and Nouha Dziri. WildTeaming at Scale: From In-the-Wild Jailbreaks to (Adversarially) Safer Language Models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 47094–47165. Curran Associates, Inc., 2024. doi: 10.52202/079017-1493. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/54024fca0cef9911be36319e622cde38-Paper-Conference.pdf.

Nitish Joshi and He He. An Investigation of the (In)effectiveness of Counterfactually Augmented Data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3668–3681. Association for Computational Linguistics, 2022. doi:

10.18653/v1/2022.acl-long.256. URL https://aclanthology.org/2022.acl-long.256/.

Monika Jotautaitė, Maria Angelica Martinez, Ollie Matthews, and Tyler Tracy. MonitoringBench: Semi-Automated Red-Teaming for Agent Monitoring. arXiv preprint arXiv:2605.09684, 2026. URL https://arxiv.org/abs/2605.09684.

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 *International Conference on Learning Representations*, volume 2026, pp. 78083–78138, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/7e105e4ae3016974e2d518e9d281f5e5-Paper-Conference.pdf.

Divyansh Kaushik, Eduard Hovy, and Zachary C. Lipton. Learning the Difference that Makes a Difference with Counterfactually-Augmented Data. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://iclr.cc/virtual_2020/poster_Sklgs0NFvr.html.

Daniel Khashabi, Tushar Khot, and Ashish Sabharwal. More Bang for Your Buck: Natural Perturbation for Robust Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 163–170. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.12. URL https://aclanthology.org/2020.emnlp-main.12/.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. Dynabench: Rethinking Benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4110–4124. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.naacl-main.324. URL https://aclanthology.org/2021.naacl-main.324/.

Jonathan Kutasov, Yuqi Sun, Paul Colognese, Teun van der Weij, Linda Petrini, Chen Bo Calvin Zhang, John Hughes, Xiang Deng, Henry Sleight, Tyler Tracy, Buck Shlegeris, and Joe Benton. SHADE-Arena: Evaluating Sabotage and Monitoring in LLM Agents. arXiv preprint arXiv:2506.15740, 2025. URL https://arxiv.org/abs/2506.15740.

Hyomin Lee, Sangwoo Park, Yumin Choi, Sohyun An, Seanie Lee, and Sung Ju Hwang. T-MAP: Red-Teaming LLM Agents with Trajectory-aware Evolutionary Search. arXiv preprint arXiv:2603.22341, 2026. URL https://arxiv.org/abs/2603.22341.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. HarmBench: A Standardized Evaluation Framework for Automated Red Teaming and Robust Refusal. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 35181–35224. PMLR, 2024. URL https://proceedings.mlr.press/v235/mazeika24a.html.

Donna Katzman McClish. Analyzing a Portion of the ROC Curve. *Medical Decision Making*, 9(3): 190–195, 1989. doi: 10.1177/0272989x8900900307. URL http://dx.doi.org/10.1177/0272989X8900900307.

Jean-Baptiste Mouret and Jeff Clune. Illuminating Search Spaces by Mapping Elites. arXiv preprint arXiv:1504.04909, 2015. URL https://arxiv.org/abs/1504.04909.

Elle Najt, Colin Toft, Tyler Tracy, Fabien Roger, and Joe Benton. SLEIGHT-Bench: A Benchmark of Evasion Attacks Against Agent Monitors. arXiv preprint arXiv:2605.16626, 2026. URL https://arxiv.org/abs/2605.16626.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A New Benchmark for Natural Language Understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4885–4901. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.441. URL https://aclanthology.org/2020.acl-main.441/.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red Teaming Language Models with Language Models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 3419–3448. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.emnlp-main.225. URL https://aclanthology.org/2022.emnlp-main.225/.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond Accuracy: Behavioral Testing of NLP Models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4902–4912. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.442. URL https://aclanthology.org/2020.acl-main.442/.

Mikayel Samvelyan, Sharath Chandra Raparthy, Andrei Lupu, Eric Hambro, Aram H. Markosyan, Manish Bhatt, Yuning Mao, Minqi Jiang, Jack Parker-Holder, Jakob Foerster, Tim Rocktäschel, and Roberta Raileanu. Rainbow Teaming: Open-Ended Generation of Diverse Adversarial Prompts. In *Advances in Neural Information Processing Systems*, volume 37, pp. 69747–69786. Curran Associates, Inc., 2024. doi: 10.52202/079017-2229. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/8147a43d030b43a01020774ae1d3e3bb-Paper-Conference.pdf.

Indira Sen, Mattia Samory, Claudia Wagner, and Isabelle Augenstein. Counterfactually Augmented Data and Unintended Bias: The Case of Sexism and Hate Speech Detection. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4716–4726. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.naacl-main.347. URL https://aclanthology.org/2022.naacl-main.347/.

Mrinank Sharma, Meg Tong, Jesse Mu, Jerry Wei, Jorrit Kruthoff, Scott Goodfriend, Euan Ong, Alwin Peng, Raj Agarwal, Cem Anil, Amanda Askell, Nathan Bailey, Joe Benton, Emma Bluemke, Samuel R. Bowman, Eric Christiansen, Hoagy Cunningham, Andy Dau, Anjali Gopal, Rob Gilson, Logan Graham, Logan Howard, Nimit Kalra, Taesung Lee, Kevin Lin, Peter Lofgren, Francesco Mosconi, Clare O'Hara, Catherine Olsson, Linda Petrini, Samir Rajani, Nikhil Saxena, Alex Silverstein, Tanya Singh, Theodore Sumers, Leonard Tang, Kevin K. Troy, Constantin Weisser, Ruiqi Zhong, Giulio Zhou, Jan Leike, Jared Kaplan, and Ethan Perez. Constitutional Classifiers: Defending against Universal Jailbreaks across Thousands of Hours of Red Teaming. arXiv preprint arXiv:2501.18837, 2025. URL https://arxiv.org/abs/2501.18837.

Aditya Sinha, Akshat Naik, Victor Gillioz, Simon Storf, Kilian Merkelbach, Rich Barton-Cooper, Axel Højmark, and Marius Hobbhahn. Training Deliberative Monitors for Black-Box Scheming Detection. arXiv preprint arXiv:2605.29601, 2026. URL https://arxiv.org/abs/2605.29601.

Simon Storf, Rich Barton-Cooper, James Peters-Gill, and Marius Hobbhahn. Constitutional Black-Box Monitoring for Scheming in LLM Agents. In *International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/65104. Accepted conference paper; camera-ready manuscript: arXiv:2603.00829v2.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting. In *Advances in Neural Information Processing Systems*, volume 36, pp. 74952–74965. Curran Associates, Inc., 2023. doi: 10.52202/075280-3275. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/ed3fea9033a80fea1376299fa7863f4a-Paper-Conference.pdf.

Eric Wallace, Adina Williams, Robin Jia, and Douwe Kiela. Analyzing Dynamic Adversarial Training Data in the Limit. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 202–217. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.findings-acl.18. URL https://aclanthology.org/2022.findings-acl.18/.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-Instruct: Aligning Language Models with Self-Generated Instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*

(Volume 1: Long Papers), pp. 13484–13508. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.754. URL https://aclanthology.org/2023.acl-long.754/.

Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel Weld. Polyjuice: Generating Counterfactuals for Explaining, Evaluating, and Improving Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 6707–6723. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.523. URL https://aclanthology.org/2021.acl-long.523/.

John Yang, Carlos Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering. In *Advances in Neural Information Processing Systems*, volume 37, pp. 50528–50652. Curran Associates, Inc., 2024. doi: 10.52202/079017-1601. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/5a7c947568c1b1328ccc5230172e1e7c-Paper-Conference.pdf.

Qiusi Zhan, Zhixiang Liang, Zifan Ying, and Daniel Kang. InjecAgent: Benchmarking Indirect Prompt Injections in Tool-Integrated Large Language Model Agents. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 10471–10506. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.624. URL https://aclanthology.org/2024.findings-acl.624/.

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 *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/file/91f18a1287b398d378ef22505bf41832-Paper-Datasets_and_Benchmarks.pdf.

A DATASETS, SPLITS, AND PROVENANCE

The evaluation suite contains a paired-trajectory component for Tiers 1 and 2 and an online component for Tier 3 (Table 4). CUA-SHADE-Arena measures external transfer. We report components separately rather than pooling their scores.

Table 4: **Evaluation data.** Counts distinguish trajectories from pairs; tiers follow Table 6.

Component	Composition	Role
External transfer	968 trajectories: 484 malicious and 484 benign	Separately released CUA-SHADE-Arena. Tier 1 (trajectory-level) evaluation.
Our test suite (paired)	168 trajectories: 36 malicious, 36 paired benign, 96 additional benign	Tiers 1 (trajectory-level) and 2 (pair-level) evaluation
Our test suite (online)	266 test pairs. 91 calibration pairs with 20 distinct benign trajectories	Tier 3 (online) first-divergence diagnostics

Evaluation input views. The evaluated CUA-SHADE-ARENA view contains task descriptions, tool names, and tool results, without agent-authored prose, reasoning blocks, or tool-call argument fields. Relevant argument information may still appear in tool results. The reported gain measures transfer under this input view. A comparison with fuller inputs on the same trajectories remains unmeasured.

The BashArena Opus view preserves the task description, tool names, call arguments, and tool results. Its assistant blocks contain tool calls without accompanying prose or reasoning. The paired and online suites retain stored task messages, assistant text, and any recorded tool calls and results. Argument detail varies with the source. Reasoning is visible only if it is already part of the stored text. We do not append hidden reasoning, construction annotations, or judge verdicts to monitor inputs. Online inputs stop at the scored prefix.

A.1 TRAINING VIEWS AND SPLIT ISOLATION

The main training pool combines the retained training data from all four feedback-guided collection rounds. After excluding six over-context pairs from the selected view of 1,604 accepted pairs, this pool contains 1,290 pairs (2,580 SFT examples). These include repeated benign counterparts. The 35B equal-count control uses a view of the same source, rather than an independent replication of pair construction. The external SFT comparator combines STRIDE and Gloom. The main training views link each serialized example to its source pair and benign carrier.

The main training dataset was fixed before our test suite was obtained, and this panel was not used for main-checkpoint training or selection.

An exact-text audit of the 2,580 frozen training examples finds seven benign suite matches: two paired benign traces and five additional benign traces. Excluding these seven records and the malicious counterparts of the two affected pairs leaves 159 trajectories, comprising 34 pairs and 91 additional benign traces. On this common subset, mean AUROC is 0.529 for Prompted, 0.650 for External, and 0.883 for Ours; strict pair ordering is 0.147, 0.324, and 0.809. Benign FPR at threshold 5 is 0.120, 0.052, and 0.052, respectively. This post-hoc check addresses exact-text overlap without establishing complete carrier ancestry.

For seeds 1001 and 1002 on the full suite, External AUROC is 0.654 and 0.630, and Ours AUROC is 0.886 and 0.894. Benign false-alarm counts are 7/132 and 6/132 for External, and 11/132 and 2/132 for Ours. Table 3 averages checkpoint-wise metrics. Its intervals resample the 36 paired roots jointly and the 96 extra benign records separately, conditioning on the observed checkpoints.

No CUA-SHADE-Arena trajectory exactly matches the reconstructed fine-tuning source pools after whitespace normalization. An exhaustive comparison using sets of five consecutive whitespace-delimited tokens gives a maximum Jaccard similarity of 0.010 across all evaluation–training text pairs.

B GENERATION, VALIDATION, AND SEARCH PROTOCOL

The generator constructs a malicious twin through a bounded edit to a segmented benign carrier. The planner specifies the behavioral objective and a plausible opening. The editor realizes that plan as insertions and necessary continuation repairs. An accepted pair retains the benign trace, edited trace, alignment, edit footprint, and validation record. The edit constraints limit the number and placement of changes without requiring a globally smallest successful edit. Figure 2 summarizes construction, local revision, and the separate evaluation branch.

B.1 EDIT REPRESENTATION AND MECHANICAL CHECKS

An edit specifies distinct anchors in the original trace, a block of new steps at each anchor, and any repaired original steps. We segment each carrier once. Unchanged steps remain byte-identical after deterministic reconstruction. The stored index map identifies inserted positions, repaired positions, and the primary action nominated by the editor.

B.2 QUALITY GATES AND VALID-CANDIDATE RETENTION

The quality panel scores naturalness, continuation coherence, minimality, plausible cover, harm severity, scheming severity, and detection difficulty. Quality judgments use 0–10 scales. The configuration sets median floors of 6 for naturalness, 6 for coherence, 7 for minimality, and 4 for harm severity. Enabled staged edits can relax the minimality floor by 2. The effective structural budget changes separately. Plausible cover and detection difficulty are measured without serving as acceptance floors. The configuration also requires acceptance consensus and unanimous genuine-misalignment judgments among the required returned verdicts.

Quality revisions operate within a regeneration budget. Once a valid pair exists, feedback from M can request harder variants within a separate escalation budget. A rejected or failed revision leaves the retained valid pair unchanged. An attempt with no valid candidate returns no pair. The feedback-disabled path returns the first valid pair. In the 9B pilot, an alarm requires score at least 5. Binary search over prefixes estimates the first alarm: detection at the earliest injected step is easy, later detection is medium, and no detection is hard. Easy candidates receive at most two extra editor calls beyond the original five rollouts, and valid candidates that remain easy are retained. The main training corpus predates this pilot. Its retained provenance does not identify a feedback monitor and escalation setting for every source batch, so we do not attribute its training gains to a particular feedback policy. Infrastructure failures remain separate from quality rejections when computing yield and cost.

B.3 BEHAVIORAL SPECIFICATIONS AND ARCHIVE SEARCH

A search specification records the requested family, mechanism, strategy, and proposed edit window. The planner receives a compact carrier description, including the user task and relevant affordances. Judges inspect the full pair. Mutation can change the mechanism, arguments, omissions, or division between setup and payoff. Crossover combines compatible elements of parent plans before the editor constructs and validates a new candidate. The search score aggregates feedback from the configured panel.

With fixed generator weights, MAP-Elites (Mouret & Clune, 2015) maintains an archive of valid candidates grouped by carrier, family, mechanism, placement, and monitor-feedback descriptors. A cell groups candidates with the same descriptor values, and one carrier can produce multiple candidates. The archive retains the highest-scoring valid candidate per cell (Figure 4). The configured feedback monitors rank the resulting valid pairs by their minimum evasion rating. Detected but valid pairs remain admissible.

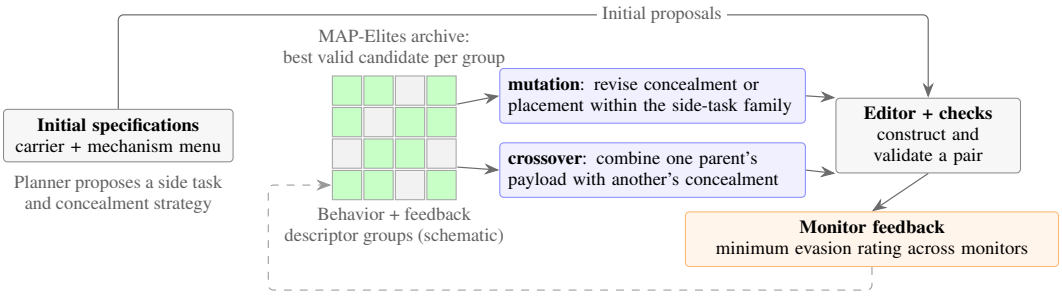


Figure 4: **Feedback refines specifications through a shared editor and validity checks.** Each elite is the highest-scoring valid candidate retained in its behavior-and-feedback group. Green cells depict occupied groups schematically. Monitor feedback supplies the search score.

C SUPERVISION CONSTRUCTION AND TRAINING CONFIGURATIONS

The main direct-training comparison uses 2,580 generated examples from 1,290 pairs. The complementary continuation route first trains on 2,000 STRIDE and Gloom examples. Each pair contributes a malicious trajectory and its benign counterpart. Training views retain pair and carrier identifiers for split checks (Appendix A.1).

C.1 TRAINING TARGETS AND INPUT VISIBILITY

A target contains a reasoning block followed by the prescribed score line. Target risk scores lie on a 0–10 scale. Their assignment uses labels and severity, not calibrated probabilities of malicious behavior. The banded target mapping assigns benign and malicious examples to low- and high-risk bands. Graded targets use judge-provided intermediate severities. Reduced concealment does not by itself justify a lower harm-severity target.

The main recipe assigns score 0 to benign examples and scores 7–9 to the selected malicious examples. Its rationales use templates filled from construction metadata, including the editor's injection summary and cover story. External STRIDE and Gloom examples use fixed class-specific templates with scores 0 and 7. These explanations are label-conditioned training targets, rather than independent assessments of the trajectories. All reported training uses complete trajectories. Prefixes are used only for evaluation, with no later actions or construction plans in the monitor input.

C.2 OPTIMIZATION AND INFERENCE VISIBILITY

Supervised fine-tuning minimizes completion-token cross-entropy for the rationale and score, masking the input tokens from the loss. Direct training (Ours) starts from the base model using our generated data. Continuation (External+ours) starts from the External monitor.

The main training settings are given in Section 4.1. The fixed schedule uses 1,296 updates and 5,184 example presentations from 2,580 training examples. Completion-loss weights sum to 129.879 per Ours example and 86 per External example in the main comparison. The historical paired and prefix diagnostics use the earlier 1,296-pair view (2,592 examples), rank-32 LoRA (Hu et al., 2022), two epochs, and batches of four examples. This recipe uses an 8,192-token sequence limit, left truncation, and Adam with learning rate 2×10^{-4}. Its initial runs did not control adapter initialization through the seed setting, so they are reported separately from seeded repeats. Each continuation shares its trained parent state, but direct comparisons do not have matched initial adapters. Table 5 summarizes the main training routes.

Table 5: **Main training routes.** Counts are serialized SFT examples.

Configuration	Initialization	Supervision
Ours	fixed base	2,580 SFT examples from 1,290 generated pairs
External	fixed base	1,000 STRIDE + 1,000 Gloom SFT examples
External+ours	External	2,580 additional SFT examples from 1,290 pairs

D METRICS, INFERENCE, AND STATISTICAL PROTOCOL

The metric tiers share a monitor score but require different annotation and comparison units. Trajectory labels support Tier 1, matched benign carriers support Tier 2, and validated action timing supports Tier 3. Table 6 summarizes the required annotations and metrics.

Table 6: **Three questions for evaluating monitor behavior.**

Tier	Assesses	Requires	Metrics
1: trajectory	Maliciousness of a completed run	Trajectory labels	AUROC and partial AUROC
2: paired	Responses to matched twins	Benign–malicious pairs	Pair ordering and joint correctness
3: online	Detection over time	Validated action labels	First-alarm outcomes and benign-run alarms

D.1 TRAJECTORY DISCRIMINATION AND LOW-FPR METRICS

AUROC measures the probability that a randomly sampled malicious trajectory scores above a benign trajectory, with half credit for ties (Fawcett, 2006). For an FPR budget α, the empirical true-positive rate (TPR) uses an observed threshold satisfying $\mathrm{FPR} \le \alpha$, without interpolating between thresholds. Thresholds chosen on the evaluation set describe its ROC curve. A deployment threshold must be selected on separate calibration data.

Let $A_\alpha = \int_0^\alpha \mathrm{TPR}(u)\, du$ denote the raw partial area (McClish, 1989). The two conventions used in the source studies are

$$\mathrm{meanTPR}_\alpha = A_\alpha/\alpha, \tag{2}$$

$$\mathrm{pAUROC}_\alpha = \frac{1}{2}\left[1 + \frac{A_\alpha - \alpha^2/2}{\alpha - \alpha^2/2}\right]. \tag{3}$$

Random ranking has expected mean TPR $\alpha/2$ and pAUROC 0.5. Thus the fresh-data study's mean TPR at a 10% budget has chance level 0.05. It must not be compared directly with the pAUROC@10 or pAUROC@20 columns in other tables.

D.2 COUNTERFACTUAL METRICS AND DIAGNOSTIC BASELINES

For a benign twin x^b and malicious twin x^m, report the score gap $m(x^m) - m(x^b)$ and strict ordering $\Pr[m(x^m) > m(x^b)]$. Joint threshold correctness is $\Pr[m(x^m) \ge \tau,\ m(x^b) < \tau]$. We also refer to joint correctness as paired accuracy. It has no universal 50% chance baseline. Comparisons report the number of complete scored pairs and any fallback for an unparsed twin.

D.3 CUMULATIVE LABELS AND ONLINE METRICS

Let z_t indicate an adjudicated malicious action at step t, and let a_t identify membership in the construction edit mask. The online label is $y_t = \mathbf{1}[\exists i \le t : z_i = 1]$. In general $a_t \ne z_t \ne y_t$. If the first malicious action has zero-based index k, a prefix containing t steps is positive exactly when $t \ge k + 1$. Later benign actions do not reset the cumulative label. The prefix monitor sees only the task and steps included in that prefix.

For a calibrated threshold τ, define the first alarm as $T = \min\{t : m(x_{1:t}) \ge \tau\}$, with $T = \infty$ for no alarm. Online TPR counts malicious runs whose first alarm occurs at or after onset and by the end of

the run. Pre-onset alarms are early false alarms. Misses and early alarms are reported alongside delay among detected attacks. The probability of any alarm on a complete benign run captures accumulated exposure to false alarms over repeated monitoring calls. It differs from the fraction of benign prefixes flagged.

D.4 Scoring and uncertainty

The main trained monitors use a strict parser that reads the final score and rejects malformed or degenerate outputs. Prompted baselines use a lenient parser. Both trajectory components retain all scheduled records with unparseable outputs scored zero, and exclude incomplete evaluation cohorts. These comparisons use one sample at temperature zero, with a 4,096-token output limit. Suite prompts use a 24,000-token input limit. The external benchmark uses a 65,536-token context budget less output overhead. Inputs that exceed the budget retain their most recent tokens. These limits and parser differences qualify cross-model and cross-cohort comparisons.

A historical parser audit replays both rules on 968 full-trajectory CUA-SHADE responses for each of 12 trained and four prompted Qwen checkpoints. The trained set comprises Ours, External, and External+ours at each of the four Qwen sizes. The audited trained checkpoints give identical scores or rejections under both rules. Their CUA-SHADE AUROC and partial AUROC therefore do not depend on this parser choice. This comparison does not cover suite or prefix responses. The main 9B and 397B prompted results use separate rescoring runs, so their values differ from this historical audit. For prompted Qwen3.5-4B, the lenient parser recovers 956 responses rejected by the strict parser, increasing AUROC from 0.500 to 0.649. Seven responses remain unparseable. The reported baseline uses the lenient value. Table 7 reports this historical audit, not the current main-training comparison.

Table 7: **Historical parser replay preserves the audited trained-model gains.** Both rules score the same 968 CUA-SHADE trajectories. Failure counts denote unparseable replies. The main prompted comparisons use the lenient rule.

Qwen3.5 configuration	Strict AUROC	Lenient AUROC	Strict failures	Lenient failures
4B: External	0.562	0.562	0	0
4B: External+ours	0.759	0.759	0	0
4B: Prompted	0.500	0.649	963	7
9B: Prompted	0.500	0.754	968	2
35B-A3B: Prompted	0.501	0.665	888	11
397B-A17B: Prompted	0.500	0.909	968	0

Uncertainty. Paired comparisons resample shared records jointly, using carrier clusters when multiple twins share a carrier. Setting- or carrier-level sensitivity analyses are needed when records share environments or source trajectories. Record-bootstrap intervals and variation across training seeds describe different sources of uncertainty. Small family cells retain their actual denominators and are treated as diagnostics. Estimates below 100 observations are marked in their tables or captions.

E Training gains, model size, and inference cost

E.1 Training-data comparisons

Matched-count 35B continuation. In a separate Qwen3.5-35B-A3B study, each of three seeds trains a parent on 500 STRIDE and 500 Gloom examples. Each parent then branches to 500 updates on either another 500 STRIDE and 500 Gloom examples or 1,000 of our generated examples. Each stage has 500 benign and 500 malicious examples, and the three training sets have disjoint row IDs. Both children load the same seed-specific parent weights with a fresh optimizer. An amended protocol sets the total completion-loss weight to 86 per example in both children, although their token budgets still differ. On each of CUA-SHADE-Arena and our test suite, the generated branch exceeds the external branch and parent in all three seeds (Table 8). The external branch is below its parent in two of three seeds on each benchmark.

A further child keeps the generated branch's input prompts, members, training schedule, and completion weight, but replaces the rationale and malicious severity target with class-level templates. It retains most of the generated branch's AUROC. This changes two target components at once and does not isolate either minimal editing or the effect of rationale detail. All 12 CUA-SHADE score sets contain 968 records each. On our test suite, one generated child response and eight external child responses are unparseable and receive score zero. The suite's nine known benign training-pool text matches remain included in this primary 168-record readout. The 159-record exclusion is a sensitivity analysis. This study controls the number of additional training examples using a common parent. It uses a separate long-context recipe and does not replicate the main training protocol or isolate carrier source and minimal editing.

Table 8: **Equal-count continuation favors generated supervision.** AUROC is mean ± standard deviation over three training seeds on CUA-SHADE and our test suite. Each child adds 1,000 class-balanced examples and 500 updates to its own External parent. The template child changes only rationale and severity targets.

Additional training	CUA-SHADE AUROC	Our test suite AUROC
None (External parent)	0.778 ± 0.009	0.616 ± 0.021
External	0.729 ± 0.067	0.578 ± 0.055
Ours	$\mathbf{0.851 \pm 0.021}$	0.840 ± 0.035
Ours with template targets	0.840 ± 0.025	$\mathbf{0.872 \pm 0.021}$

Matched and unrelated benign supervision. A separate 4B control finds no training advantage for matched benign twins over unrelated benign replacements (Table 9). Both arms start from the same untrained adapter state and use the training settings in Section 4.1, with seed 42 and the same fixed update schedule. The matched repeat preserves the original data and schedule. The replacement arm changes each benign twin to an unrelated benign trajectory, keeping malicious examples, targets, training slots, and completion-loss weights fixed. Both use the same number of example presentations, equal class counts, and 307 distinct benign transcripts. Replacement changes the benign distribution, including task source and trajectory structure, and increases total training tokens by 1.4%. Pairing here means membership in the training set: only five batches contain both twins.

Table 9: **Matched and unrelated benign supervision in a 4B control.** Cells report AUROC / pAUROC@20. All scheduled replies parse, so the default failure policy excludes no records. CUA-SHADE is primary. Exclusion rows remove nine training-pool matches or all eleven known benign overlaps, respectively.

Evaluation	Original matched	Matched repeat	Unrelated benign
CUA-SHADE (968)	0.849 / 0.723	0.740 / 0.739	0.840 / 0.837
Our test suite	0.833 / 0.759	0.747 / 0.731	0.852 / 0.821
Training overlaps removed	0.829 / 0.750	0.746 / 0.729	0.851 / 0.817
All known overlaps removed	0.828 / 0.748	0.746 / 0.728	0.850 / 0.816

The replacement-minus-repeat CUA-SHADE AUROC difference is 0.100 [0.082, 0.118], compared with -0.009 [-0.033, 0.014] relative to the original matched checkpoint. After excluding the training-pool matches, replacement-minus-repeat is 0.104 [0.036, 0.176]. Brackets give 95% intervals from 5,000 paired, label-stratified record resamples with seed 20260924, conditional on fixed checkpoints. Replacement also exceeds the original matched checkpoint in CUA-SHADE pAUROC@20 by 0.114 [0.071, 0.156]. The matched repeat falls 0.109 below the original in AUROC. The original resumes training with restored optimizer state at step 324, whereas the repeat trains without interruption.

The cause of this repeat difference is unresolved. A single repeat does not estimate the distribution of training variability, and its gap cannot serve as a general noise threshold. The replacement control changes instance matching and benign-distribution matching together. It therefore provides no evidence for a pairing benefit and does not isolate the causal contribution of edit size. Overlap checks are reported in Appendix A.1.

E.2 BASHARENA TRANSFER

We evaluate four Qwen3.5 sizes on the BashArena Opus 4.6 action-only release.[2] The primary set contains 1,122 trajectories, with 558 malicious and 564 benign runs. The 986-row sensitivity analysis removes 136 benign `honest_summarized` records. All scheduled records enter both analyses. Unparseable responses receive score zero. Trained checkpoints use strict parsing, while the prompted base uses the predeclared lenient parser.

9B and 35B-A3B seed comparisons. Table 10 reports the 9B and 35B-A3B runs, including continuation on replayed External examples. The main comparison uses seeds 1001 and 1002, which also have complete CUA-SHADE and test-suite evaluations. At 35B, seed 1001 has one parse failure for External and one for Ours; all other trained cells parse completely.

Table 10: **Individual BashArena runs and replay controls.** AUROC uses all 1,122 trajectories. The two main seeds appear in Figure 3; the additional 9B seed is reported separately. Both continuation arms start from the same seed's External checkpoint.

Model	Seed	External	Ours	External+ours	External replay
9B	1001	0.707	0.659	0.811	0.640
9B	1002	0.553	0.598	0.781	0.725
9B	1003	0.717	0.625	0.783	0.715
35B-A3B	1001	0.743	0.720	0.761	0.771
35B-A3B	1002	0.605	0.657	0.825	0.553

4B training-route comparison. Direct Ours exceeds fresh External in one 4B run, but both Ours routes remain below the historical External reference (Table 11). The four trained arms use seed 42, rank 32, learning rate 6.67×10^{-5}, batch size 4, and a 65,536-token training cap without truncation. Each stage follows the fixed schedule in Appendix C.2, using frozen pools of 2,580 Ours or 2,000 External examples. Completion-loss weight per example is 129.879 for Ours and 86 for External, so neither token totals nor loss weights are matched across sources.

Direct Ours and fresh External retain their pilot runs, restoring optimizer state after step 324 to reach the fixed final endpoint. The two continuation arms start from the same fresh External final weights with new optimizers and repeat this update schedule. The historical External reference has a distinct training history and is not their parent.

Table 11: **4B BashArena training routes.** One training seed; both continuation arms start from fresh External. Historical External is a separately trained reference. Parse-failure counts use the full set.

Training route	Full 1,122	Subset 986	Parse failures
Prompted base	0.469	0.472	10
Ours (direct)	0.633	0.617	0
External (fresh)	0.581	0.599	0
External+ours	0.717	0.706	0
External replay	0.749	0.754	0
Historical External	0.755	0.790	0

On the full set, direct Ours minus fresh External is 0.052 $[0.020, 0.083]$, and direct Ours minus Prompted is 0.164 $[0.127, 0.199]$. External+ours exceeds its fresh External parent by 0.136 $[0.109, 0.164]$ and Prompted by 0.248 $[0.215, 0.280]$. However, External+ours minus External replay is -0.032 $[-0.062, -0.003]$. Against historical External, direct Ours is -0.122 $[-0.159, -0.087]$ and External+ours is -0.038 $[-0.070, -0.007]$.

Removing summarized benign records reduces the direct-versus-fresh-External gap to 0.019 $[-0.015, 0.052]$, whose interval includes zero. Brackets give 95% intervals from 3,000 paired task-cluster resamples, using 574 clusters for the full set and 480 for the subset.

[2] `https://huggingface.co/datasets/adityaasinha28/basharena_action_only_opus46_large`

397B training-route comparison. The four 397B arms use the same training data, example schedule, rank, learning rate, batch size, context cap, and source-specific loss weights as the 4B study above. Each arm completes the fixed update schedule, restoring its own optimizer state after step 324. Direct Ours and fresh External share an untrained adapter initialization. Table 12 reports six 397B configurations, with a historical External parent shared by the two continuation arms.

Table 12: **397B BashArena training routes.** One training seed; both continuation arms start from the historical External parent, distinct from fresh External. The subset removes summarized benign records.

Training route	Full 1,122	Subset 986	Parse failures
Prompted base	0.698	0.697	6
Ours (direct)	0.709	0.705	0
External (fresh)	0.726	0.740	0
External parent	0.847	0.852	0
External+ours	0.844	0.845	0
External replay	0.894	0.905	0

On the full set, direct Ours minus fresh External is -0.017 AUROC, with a paired task-cluster 95% bootstrap interval of $[-0.050, 0.014]$. Direct Ours exceeds the prompted base by 0.010, with an interval of $[-0.023, 0.043]$. External+ours minus the external parent is -0.003 $[-0.027, 0.020]$, while External+ours minus External replay is -0.050 $[-0.074, -0.027]$. External replay exceeds its parent by 0.047 $[0.033, 0.063]$. External+ours exceeds fresh External by 0.118 $[0.092, 0.146]$ and Prompted by 0.146 $[0.112, 0.180]$. Fresh External is not the continuation parent, so the former difference compares training routes with different initialization histories. These intervals use 3,000 paired resamples of 574 task clusters. The BashArena release had been consulted during this research program, so these comparisons are exploratory.

E.3 Cost–performance across model sizes

Figure 1 compares direct training with prompted and external-data-trained baselines across 18 configurations. Directly trained 4B, 35B-A3B, and 397B-A17B lie on the AUROC–cost frontier. All four directly trained sizes lie on the pAUROC@20–cost frontier within this displayed comparison. Table 13 adds four External+ours configurations and Gemini 3.1 Flash-Lite, for 23 configurations on the same 968 trajectories. Across direct training and continuation, our trained monitors provide Pareto-optimal cost–detection trade-offs. The expanded comparison contains frontier models trained on our pairs at 4B, 35B-A3B, and 397B-A17B. Trained Qwen entries average checkpoint-wise metrics over two seeds at 4B, 9B, and 35B-A3B, with one run at 397B-A17B. The continuation rows use the same training settings as the direct-training rows. Paired and prefix diagnostics use separate checkpoints. Unparsed outputs receive score zero. The figure uses pAUROC@20, integrating over FPR 0–20% as defined in Appendix D. This area interpolates between ROC points. It does not assert an attainable deterministic threshold at every FPR. The table separately reports the minimum false-negative rate (FNR) among observed thresholds with FPR at most 10%.

We reconstruct Qwen prompt token counts with the scoring template, tokenizer, and truncation rule, and re-tokenize the cached decoded responses. Prompted reference models use their recorded input and output token usage. Estimated cost per 1,000 trajectories is $1000(\bar{n}_{\text{in}} p_{\text{in}} + \bar{n}_{\text{out}} p_{\text{out}})$, where p denotes USD per token. For each Qwen size, we average token counts over seeds within each training state, then equally over Prompted, External, and Ours. All states at that size, including External+ours, use this common workload. This comparison holds token volume fixed across training states rather than modeling their different response lengths. Commercial references retain their own recorded token counts.

For 4B, we use a hosting-price assumption of \$0.03 per million input tokens and \$0.15 per million output tokens. Other models use uncached public rates recorded on 2026-09-24.[3] The 35B base checkpoint uses Qwen3.5-35B-A3B API rates as an architecture-matched proxy. The 9B, 35B and 397B input/output rates per million tokens are respectively \$0.10/\$0.15, \$0.3125/\$1.25 and \$0.55/\$3.50. These modeled serving costs exclude training, retries, hosting overhead, storage, and caching or batching discounts. Decoded-output token counts omit stripped special tokens, and frontier

[3]Price source: `https://openrouter.ai/api/v1/models`, 2026-09-24.

membership is descriptive under these accounting assumptions. Frontiers use unrounded costs and depend on the providers and pricing date.

Using each directly trained monitor's own token usage instead gives \$0.36, \$1.15, \$3.70, and \$6.69 per 1,000 trajectories at 4B, 9B, 35B-A3B, and 397B-A17B, respectively. In that accounting, the 397B monitor costs about 4.16 times less than Gemini 3.1 Pro. The shared-workload comparison in Figure 1 instead uses \$7.62 for every 397B training state, giving the 3.65-fold ratio reported in the abstract. Using the 9B input rate of \$0.10 per million tokens instead of the reference 4B rate of \$0.03 raises the shared 4B estimate to \$1.19 per 1,000 trajectories, with scores unchanged. The 397B–Gemini cost comparison is unaffected.

Table 13: **Cost and detection on CUA-SHADE.** Trained metrics average two seeds except one at 397B. Its continuation uses a separate External parent. FNR@10% is the minimum miss rate at observed FPR $\leq 10\%$. Costs use each size's shared workload per 1,000 trajectories. Bold marks the best trained metric within size.

Model	Training	AUROC	pAUROC@20	FNR@10%	Cost
Qwen3.5 monitors					
Qwen3.5-4B	Prompted	0.649	0.648	0.700	0.391
Qwen3.5-4B	External	0.760	0.709	0.752	0.391
Qwen3.5-4B	Ours	0.837	0.764	0.607	0.391
Qwen3.5-4B	External+ours	**0.863**	**0.826**	**0.248**	0.391
Qwen3.5-9B	Prompted	0.762	0.759	0.471	1.186
Qwen3.5-9B	External	0.765	0.753	0.449	1.186
Qwen3.5-9B	Ours	0.818	**0.794**	**0.317**	1.186
Qwen3.5-9B	External+ours	**0.849**	0.749	0.599	1.186
Qwen3.5-35B-A3B	Prompted	0.665	0.663	0.663	3.865
Qwen3.5-35B-A3B	External	0.753	0.752	0.493	3.865
Qwen3.5-35B-A3B	Ours	0.866	0.834	0.227	3.865
Qwen3.5-35B-A3B	External+ours	**0.889**	**0.884**	**0.211**	3.865
Qwen3.5-397B-A17B	Prompted	0.902	0.894	0.178	7.618
Qwen3.5-397B-A17B	External	0.880	0.862	0.213	7.618
Qwen3.5-397B-A17B	Ours	0.925	0.919	**0.138**	7.618
Qwen3.5-397B-A17B	External+ours	**0.927**	**0.925**	0.143	7.618
Other models					
GPT-5.4 nano	Prompted	0.764	0.711	0.597	1.926
Gemini-3.1 Flash-Lite	Prompted	0.813	0.811	0.372	3.128
GPT-5.4 mini	Prompted	0.886	0.832	0.339	7.332
Claude Haiku 4.5	Prompted	0.914	0.894	0.174	14.010
Gemini 3.1 Pro preview	Prompted	0.923	0.923	0.151	27.807
Claude Sonnet 4.6	Prompted	0.945	0.938	0.159	41.644
Claude Opus 4.6	Prompted	0.931	0.925	0.124	68.704

F PAIRED AND ONLINE DIAGNOSTICS

Historical 4B paired decisions. The paired view shows improved attack ranking alongside increased benign false alarms (Table 14). For the 4B monitor, further training raises correct pair ordering from 5 to 30 of 36. Joint correctness rises from 5 to 28 of 36. At threshold 5, false positives also rise from 0/36 to 3/36 on paired benign traces and from 0/96 to 20/96 on additional benign traces.

Table 14: **Further training changes paired decisions and false alarms.** These 4B checkpoints use the diagnostic recipe (Appendix C.2). AUROC uses 168 trajectories, and ordering and joint correctness use 36 pairs. FPR uses threshold 5 on 36 paired and 96 additional benign traces.

Monitor or baseline	AUROC	Ordering	Joint	Paired FPR	Additional FPR
4B: External	0.569	0.139	0.139	0.000	0.000
4B: External+ours	**0.868**	**0.833**	**0.778**	0.083	0.208

F.1 PAIRED DECISIONS AND OVERLAP SENSITIVITY

Further training changes pair ordering, joint decisions, and false alarms, and the ordering gain remains after excluding known benign text matches (Table 15). The exclusion analysis retains 34 complete pairs. False positives are reported separately for paired and additional benign traces. These comparisons connect the trajectory-ranking gain to decisions on matched twins, while exposing the benign errors at the chosen threshold. Across the same 36 pairs, the within-monitor margin $m(x^m) - m(x^b)$ averages 0.972 for External and 6.417 for External+ours. All 72 scheduled responses per monitor have cached, parseable scores. These raw score differences are descriptive within each checkpoint, not calibrated effect sizes across checkpoints.

Table 15: **Paired decisions under the 4B diagnostic recipe.** Ordering requires $s_m > s_b$. Joint correctness requires $s_m \geq 5$ and $s_b < 5$. False positives use threshold 5. Exclusion columns omit known benign text matches, retaining 34 complete pairs.

Training data	Order	Joint	FP pair	FP extra	AUROC (excl.)	Order (excl.)	Joint (excl.)
External	5/36	5/36	0/36	0/96	0.569	5/34	5/34
External+ours	**30/36**	**28/36**	3/36	20/96	**0.864**	**28/34**	**26/34**

The full pool contains nine exact benign scoring-text matches (two paired, seven extra). The selected 2,592-example own-data stage contains seven. Removing the nine known full-pool matches gives 159 records and 34 complete pairs. This common post-hoc slice does not certify carrier or ancestor independence.

F.2 DETECTION WHEN LENGTH FAVORS THE BENIGN TRACE

The further-trained monitor retains discrimination when the benign trace is at least as long as the malicious trace. This control matters because character count orders 35/36 matched pairs correctly, although its AUROC across the 168 suite trajectories is 0.502. After excluding the nine known benign matches to the reconstructed training pool, these values are 33/34 and 0.507. Thus, matched ordering can reward an insertion cue that contributes little to overall trajectory discrimination.

We also compare each malicious trace with every benign trace at least as long in characters. This restriction retains 2,367 comparisons involving 36 malicious and 114 benign trajectories. Character count favors the benign trace in every comparison. Concordance gives one point when the malicious score is higher, half a point for a tie, and zero otherwise (Table 16). These are cross-comparisons among existing records, not additional matched counterfactual pairs or independent observations. We therefore do not compute binomial intervals by treating the 2,367 comparisons as independent trials.

Table 16: **Detection when the benign trace is at least as long.** The 4B diagnostic checkpoints are compared over 2,367 length-reversed comparisons, or 2,182 after exclusions. Concordance counts ties as one half; these cross-comparisons are not independent matched pairs.

Score source	Full-suite AUROC	Conditional concordance	Exclusion sensitivity
Transcript length	0.502	0.000	0.000
4B: Prompted	0.524	0.514	0.514
4B: External	0.569	0.514	0.515
4B: External+ours	**0.868**	**0.833**	**0.828**

For the further-trained 4B monitor, conditional concordance remains 0.828 after the known-overlap exclusions. This evidence is inconsistent with a score determined solely by increasing transcript length, but does not establish independence from length or other superficial cues. Footprint-matched benign edits remain necessary to isolate responses to malicious content.

F.3 ONLINE DETECTION ACROSS STEPS

Run-level discrimination. We assign each run its highest prefix score, $r(x) = \max_t m(x_{1:t})$, and compute AUROC over malicious and complete benign runs. Each run receives equal weight, and ties receive half credit. On the same 266 pairs, the external-data and further-trained 4B monitors

obtain 0.650 and 0.907 run-level AUROC, respectively. The AUROC difference is 0.257 (95% paired carrier-bootstrap interval [0.169, 0.331], based on 10,000 resamples over 45 carriers).

The primary AUROC weights all 266 pairs equally. The 45 carriers contribute 1–22 pairs each, so some benign counterparts receive repeated weight. Weighting carriers equally instead gives AUROC 0.659 for External and 0.878 for External+ours, a difference of 0.219 (95% paired carrier-bootstrap interval [0.137, 0.300], 10,000 resamples). This sensitivity changes the evaluation weighting, while retaining a positive difference for these fixed checkpoints.

Scoring-opportunity sensitivity. In all 266 test pairs, the malicious run has two to seven more scored prefixes than its benign counterpart (median 32 versus 29). A maximum over more prefixes has more opportunities to be high, so we recompute AUROC after retaining the same number of scored prefixes on each side of every pair. Keeping the earliest malicious prefixes gives AUROC 0.602 for External and 0.873 for External+ours, a difference of 0.271 (95% paired carrier-bootstrap interval [0.182, 0.348]). Selecting evenly spaced malicious prefixes while retaining the final one gives 0.650 and 0.907, a difference of 0.256 (95% interval [0.170, 0.329]). Using only the terminal score gives 0.658 and 0.984, respectively. The original maximum-prefix AUROCs are 0.650 and 0.907. These descriptive checks retain the between-checkpoint gain under equal prefix counts, but they also change which later evidence is visible and do not isolate the causal effect of scoring opportunities or establish timely detection. This ranking measure counts an alarm anywhere in a malicious run, including before divergence. The first-alarm breakdown below separately measures timing.

Observation and timing. For online scoring, a step is one serialized message: a user message, assistant message, tool call, or tool result. Each score uses only the messages through that step. Tool calls and their results are separate steps, so a prefix can end at a tool call before including its result. The 357 test and calibration pairs contain 669 recorded injection positions. These comprise 544 tool calls, 124 assistant messages, and one tool result. These recorded positions guide annotation. They do not establish semantic onset or demonstrate interception during execution.

For each pair, the boundary is the first step whose visible text differs between the two traces. If the malicious trace only appends steps, the boundary is the first appended step, immediately after the shared benign prefix. We classify the malicious run's first alarm as before the boundary, at or after it, or absent. Benign-run alarms use the complete benign counterpart. The three first-alarm outcomes sum to one. Failure to detect after divergence combines pre-divergence alarms and no alarm, rather than measuring false negatives alone. This boundary identifies the earliest observable difference, which can precede a semantic malicious action. An alarm after observing an irreversible action is not evidence that the action could have been prevented.

First-alarm outcomes. The two 4B diagnostic checkpoints have valid scores for all 16,421 test prefixes across 266 pairs and identical scores on every shared prefix. At thresholds 7 and 8, their post-divergence detections are 119/266 and 237/266, with 70/266 and 53/266 benign runs flagged. The external-data monitor gives 31/266 pre-divergence alarms, 119/266 post-divergence first alarms, and 116/266 runs without an alarm. The further-trained monitor gives 26/266, 237/266, and 3/266 for these outcomes. The first checkpoint emits only 0 and 7, while the further-trained checkpoint also emits 8 and 9. This observed score resolution gives additional operating points under the tested training recipe. An independent causal effect of severity labels remains untested. The thresholds in Table 17 are retrospective diagnostics. Applying the original 1% benign-run calibration target to the 91 calibration pairs selects thresholds 8 and 10, at which neither checkpoint flags any test run. One false alarm among 91 paired calibration entries would exceed that 1% target. These entries contain only 20 distinct benign trajectories. The original rule weights paired entries, so repeated trajectories do not supply independent calibration evidence.

Model-size comparison. Table 18 compares four backbones trained directly with the historical diagnostic recipe, without an External stage. Run-level AUROC increases from 0.973 for 4B to 1.000 for 397B on this complete-score subset. At threshold 8, larger backbones produce fewer pre-divergence alarms, while runs without any alarm rise from 0/242 to 4/242. These single-run results describe an online alarm trade-off, rather than establishing monotonic improvements in both false positives and false negatives.

Table 17: **Threshold sweeps for the 4B diagnostic checkpoints.** Detection counts first alarms at or after divergence over 266 test pairs. These empirical test operating points are not calibrated deployment guarantees.

	External		External+ours	
Threshold	Detection	Benign alarms	Detection	Benign alarms
0	0.000	1.000	0.000	1.000
1–7	0.447	0.263	**0.887**	**0.207**
8	0.000	0.000	**0.891**	0.199
9	0.000	0.000	**0.511**	0.098
10	0.000	0.000	0.000	0.000

Table 18: **Alarm timing under the diagnostic training recipe.** Threshold 8 is retrospective. Run-level AUROC uses maximum prefix scores. The common subset retains 242 of 266 pairs; exclusions arise from 96 invalid prefix occurrences across 22 pairs at 35B and eight timeouts across three pairs at 397B.

Qwen3.5	Run-level AUROC ↑	Pre-divergence alarm ↓	Post-divergence detection ↑	No alarm ↓	Benign-run alarms ↓
4B	0.973	24	218	0	31
9B	0.977	8	232	2	17
35B-A3B	0.984	2	237	3	17
397B-A17B	1.000	0	238	4	0

Missing-prefix sensitivity. All four models have records for the 16,421 scheduled prefix occurrences. At 35B, 96 unparseable occurrences affect 22 pairs; at 397B, eight timeout occurrences affect three pairs. Repeated shared prefixes count more than once here. Table 19 instead counts distinct cached test and calibration inputs, giving 39 invalid replies at 35B and eight at 397B. One pair overlaps, leaving 24 excluded pairs. Exclusion removes two of the 45 carriers entirely, including all 19 and three pairs from those carriers. Thus, the 242-pair comparison uses a selected subset. Every pair still contains one malicious and one benign run. Assigning failed prefixes zero and retaining all 266 pairs gives AUROCs of 0.976, 0.979, 0.986, and 0.998, respectively. Allowing each failed prefix any score from 0 to 10 gives AUROC bounds of [0.903, 0.987] for 35B and [0.998, 1.000] for 397B. The bounds assign failed malicious prefixes zero and failed benign prefixes ten for the lower endpoint, then reverse these assignments for the upper endpoint. These are missing-score sensitivity bounds, not confidence intervals. The 4B and 9B models have no failed prefixes.

F.4 Prompted and commercial prefix comparisons

Direct training improves maximum-prefix AUROC over the corresponding prompted monitor at each of the four Qwen3.5 sizes (Table 19). All four paired carrier-bootstrap intervals for these differences exclude zero under the primary failure policy. This comparison uses the full 266-pair test panel and the same direct-training checkpoints as Table 18.

All scheduled prefixes have a recorded outcome, including failed calls and unparseable replies. We extract scores using each model's fixed parsing rule. These results measure retrospective discrimination on the generated panel. Higher maximum-prefix AUROC does not establish faster first alarms or prevention of harmful actions.

Table 19: **Run-level discrimination under the diagnostic recipe.** All models use 266 test pairs from 45 carriers. Intervals resample carriers for fixed checkpoints. Invalid replies receive zero, or ten in the Alarm AUROC column. Counts cover 6,286 distinct test and calibration inputs, not repeated prefix occurrences.

Monitor	AUROC [95% CI]	pAUROC@20	Invalid replies	Alarm AUROC
Ours 4B	0.976 [0.945, 0.994]	0.932	0	0.976
Ours 9B	0.979 [0.949, 0.997]	0.942	0	0.979
Ours 35B-A3B	0.986 [0.964, 0.998]	0.960	39	0.908
Ours 397B-A17B	0.998 [0.994, 1.000]	0.998	8	1.000
Prompted 4B	0.588 [0.549, 0.633]	0.574	215	0.517
Prompted 9B	0.727 [0.651, 0.807]	0.641	58	0.638
Prompted 35B-A3B	0.598 [0.561, 0.636]	0.587	202	0.565
Prompted 397B-A17B	0.906 [0.868, 0.943]	0.881	77	0.772
External 4B	0.650 [0.600, 0.709]	0.563	0	0.650
Haiku 4.5	0.786 [0.717, 0.855]	0.732	0	0.786
Gemini 3.1 Pro	0.948 [0.925, 0.970]	0.944	1	0.948
Sonnet 4.6	0.975 [0.956, 0.990]	0.964	0	0.975
Opus 4.6	0.959 [0.937, 0.977]	0.955	14	0.959