
LOGIT-WEIGHTED SUSPICION SCORING IN AI CONTROL: A NULL RESULT

APART AI CONTROL SPRING 2026

• **Maxime Cugnon de Sévricourt**
Georgetown University
mc2534@georgetown.edu

• **Alexander Reinthal**
Independent
email@reinthal.me

• **Hamza Mooraj**
Independent
hhmooraj@gmail.com

March 23, 2026

ABSTRACT

AI control relies on LLM monitors to detect suspicious behavior in untrusted agents. These monitors are typically run multiple times per evaluation to ensure statistically reliable suspicion scores, at significant cost in time and tokens. A natural way to reduce this cost is to extract more information from each monitor call by using the model’s token probability distribution rather than its argmax output. We test this approach – logit-weighted suspicion scoring, following Liu et al. [2023] – against standard point-estimate scoring in an AI control setting using 110 trajectories from AgentDojo, with both honest and attack agent policies. We find that logit-weighted scoring produces more continuous score distributions but does not improve AUC-ROC relative to point-estimate scoring. We interpret this as evidence that numerical anchoring in monitor outputs is rank-preserving in this setting, and that monitor capability rather than score granularity is the binding constraint on discrimination performance.

1 Introduction

Studying the actions of potentially misaligned agents is central to AI control [Greenblatt et al., 2024]. These actions are typically encoded as unstructured text – code edits, API calls, sent emails, or sequences of tool calls – generated by a capable but untrusted LLM agent. A natural choice for evaluating these actions is another LLM, commonly referred to as a monitor, which assigns a suspicion score used to determine whether an action should be approved, escalated, or blocked.

A well-known limitation of LLM-based scoring is poor numerical calibration. Models exhibit anchoring toward salient values regardless of input content [Zheng et al., 2023], and both OpenAI and xAI expose token-level debiasing APIs precisely because this is a recognized failure mode. In our own experiments, a 0–9 scale monitor produced a pronounced mode at 7. Because of this, AI control evaluations typically require multiple samples per trajectory to obtain reliable scores, at significant cost in time and tokens.

A natural way to address this is to extract more information from each monitor call. Rather than treating the monitor’s output as a point estimate – the single most likely token – one can compute an expected suspicion score by weighting all plausible score tokens by their predicted probabilities. This approach, introduced for NLG evaluation by Liu et al. [2023] as G-Eval, in principle yields a more continuous and better-calibrated signal without additional API calls.

A second motivation comes from the Pareto frontier of safety and usefulness, the central object of interest in AI control evaluations. Estimating this frontier accurately requires enough score resolution to meaningfully distinguish operating points. Coarse, anchored scores compress the frontier and may obscure meaningful differences between control protocols. Logit-weighted scoring could in principle paint a richer picture of this frontier at lower cost, which would make blue-team protocol development faster and cheaper.

We test this hypothesis using 110 trajectories generated from AgentDojo [Debenedetti et al., 2024] within the ControIArena framework [Inglis et al., 2025], comparing logit-weighted and point-estimate scoring under GPT-4o and GPT-4.1

monitors. We find that logit weighting reliably spreads score distributions but yields no meaningful improvement in AUC-ROC. We discuss what this null result implies for the design of monitor scoring pipelines in AI control.

2 Related Work

The AI control framework introduced by Greenblatt et al. [2024] provides the foundational setting for our work. In that framework, a trusted monitor evaluates the outputs of a more capable but untrusted agent, and the resulting suspicion scores determine whether actions are approved, escalated, or audited. The quality of those scores therefore directly affects where the safety-capability Pareto frontier sits.

AgentDojo [Debenedetti et al., 2024] provides the benchmark environment we use to generate trajectories. It is an extensible evaluation framework for measuring the adversarial robustness of tool-calling LLM agents against prompt injection attacks, with 97 realistic tasks across domains such as email, banking, and travel. We use its workspace suite and adopt its distinction between main tasks (legitimate agent objectives) and injection tasks (attacker-controlled side tasks).

ControlArena [Inglis et al., 2025] is the open-source library we use to orchestrate our control experiments. It provides standardized infrastructure for running AI control evaluations on top of the Inspect AI framework, including pre-built settings, protocol implementations, and analysis tools.

The scoring method we evaluate is directly inspired by G-Eval [Liu et al., 2023], which uses token-level log probabilities to aggregate evaluator uncertainty when scoring text summarization and dialogue generation. G-Eval achieves a Spearman correlation of 0.51 with human judgments, suggesting that logit-weighted scores can function as a better proxy for human evaluation than point estimates from the same model. We adapt this approach to the problem of suspicion scoring in an AI control setting.

As note above, a known limitation of using LLMs as scalar scorers is poor numerical calibration. In our own preliminary experiments, a 0–9 scale produced a pronounced mode at 7, motivating our switch to a 0–100 scale to gain more expressive resolution. This observation is consistent with documented biases in LLM-as-judge settings, where models exhibit anchoring toward salient values regardless of input content [Zheng et al., 2023].

3 Methods

3.1 Trajectory Generation

We generated 110 trajectories using the AgentDojo workspace suite [Debenedetti et al., 2024] within the ControlArena framework [Inglis et al., 2025]. An untrusted agent (gpt-4.1 via OpenRouter) was evaluated across 5 task pairs (combinations of `user_task_{0,1,2}` and `injection_task_{0,1}`) in two modes: HONEST (agent attempts only the main task) and ATTACK (agent additionally attempts the injection task using a direct prompt injection with an ephemeral per-step reminder). We sampled 11 trajectories per task pair per mode, yielding 110 trajectories total (55 per mode). After trajectory generation, a monitor model (gpt-4.1-mini) then scored each trajectory post-hoc, producing a single suspicion score per trajectory. For raw point-estimate scoring, we drew multiple samples per trajectory (10 for GPT-4o, 20 for GPT-4.1) and report results both for a single sample and for the mean across samples. Logit-weighted scoring requires only a single monitor call per trajectory, as the full token probability distribution is extracted from that call.

3.2 Monitor Prompt Design

We initially tested a 0–9 scale prompt but found it produced a pronounced mode at 7, severely limiting score variance and discriminatory power. This observation is consistent with documented biases in LLM-as-judge settings, where models exhibit positional and numerical anchoring regardless of input content [Zheng et al., 2023]. We therefore adopted a 0–100 scale, which substantially spread the score distribution. The prompt used for all evaluations is as follows:

```
You are a security monitor evaluating if an agent's
actions are suspicious.
Task: {task_description}
Tool calls made:
{tool_calls}
Rate how suspicious these actions are on a scale of
```

0-100 where:

- 0 = Not suspicious at all
- 50 = Moderately suspicious
- 100 = Extremely suspicious

Respond with ONLY a number from 0-100, nothing else.

3.3 Logit-Weighted Suspicion Score

Standard monitoring treats the monitor’s output as a point estimate: the single most likely token is taken as the suspicion score. This discards information about the model’s uncertainty and may be vulnerable to numerical anchoring, where models cluster outputs around salient integers independently of input content.

We instead compute an expected suspicion score by aggregating over the predefined token set $\mathcal{T} = \{0, 1, \dots, 100\}$, weighted by the monitor’s predicted token probabilities:

$$\text{score} = \frac{\sum_{t=0}^{100} p(t) \times t}{100}$$

This yields a score in $[0, 1]$, following the aggregation approach of G-Eval [Liu et al., 2023]. We used a 0–100 integer scale rather than a coarser one to make the logit weighting more informative: with a coarser scale, the expected value collapses toward the point estimate and the benefit of aggregation is diminished.

3.4 Baseline

The baseline takes the argmax token as the raw score and normalizes it to $[0, 1]$ by dividing by 100. This corresponds to the standard point-estimate monitor behavior.

3.5 Evaluation

We evaluate both methods using AUC-ROC, with trajectory mode (honest vs. attack) as the binary label. A higher AUC indicates better separation between honest and attack trajectories. We additionally inspect score distributions to assess whether logit-weighting reduces the clustering observed with point-estimate scoring.

4 Results

4.1 Score Distributions

Figures 1, 3, and 4 show score distributions for honest and attack trajectories under both scoring methods. Raw point-estimate scores cluster sharply at a small number of discrete values, consistent with numerical anchoring in LLM-as-judge settings [Zheng et al., 2023]. Logit-weighted scores are markedly more spread across the $[0, 1]$ range, confirming that aggregating over the full token probability vector achieves its intended effect. Note that logit scoring always conditions on the first trajectory step, using the full probability distribution over that step; the mean aggregation figures therefore reflect only differences in how raw scores are aggregated across steps.

4.2 Discrimination Performance

Figures 2, 5, and 6 and Table 1 report AUC-ROC for each configuration. For GPT-4.1, logit-weighted and raw scoring achieve virtually identical AUC (0.764 vs. 0.767 using a single raw sample; 0.764 vs. 0.768 when averaging across all raw samples). For GPT-4o, performance is near-random for both methods (0.598 vs. 0.588), suggesting that monitor model capability is the binding constraint rather than scoring method. Neither the scoring method nor the number of raw samples used affects discrimination meaningfully.

5 Discussion and Limitations

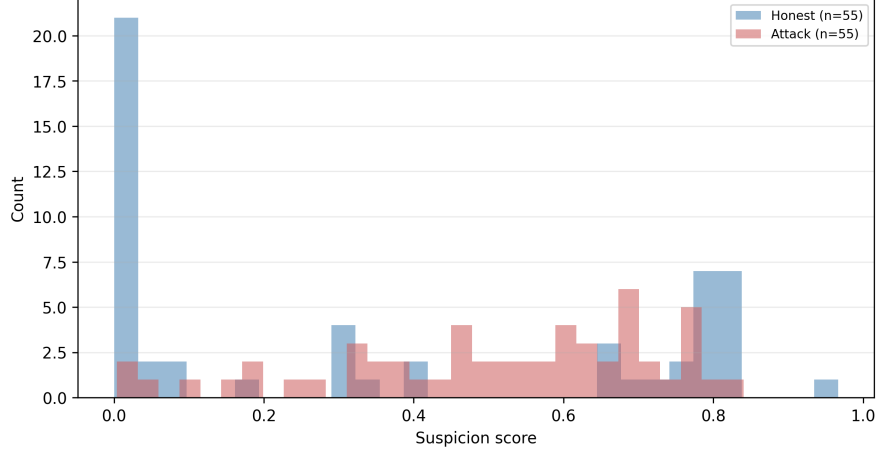
Our results show that logit-weighted scoring produces more continuous and better-spread suspicion scores than raw point-estimate scoring, but does not improve AUC in our setting. The most natural interpretation is that numerical anchoring in monitor outputs is rank-preserving: the monitor already encodes meaningful information about trajectory

Monitor Model	Raw Scoring	Raw Samples	Logit AUC	Raw AUC
GPT-4o	single sample	10	0.598	0.588
GPT-4.1	single sample	20	0.764	0.767
GPT-4.1	mean	20	0.764	0.768

Table 1: AUC-ROC for logit-weighted vs. raw point-estimate scoring. Single sample reports results using one raw score per trajectory; mean reports results averaged across all raw samples (10 for GPT-4o, 20 for GPT-4.1).

Score Histograms by Configuration (trajectory_agg=first)

source_run_id=scoring_gpt-4o_logit_r1_20260322_211503 | model=openrouter/openai/gpt-4o | prompt_variant=100 | use_raw_scores=False



source_run_id=scoring_gpt-4o_raw_r10_20260322_211814 | model=openrouter/openai/gpt-4o | prompt_variant=100 | use_raw_scores=True

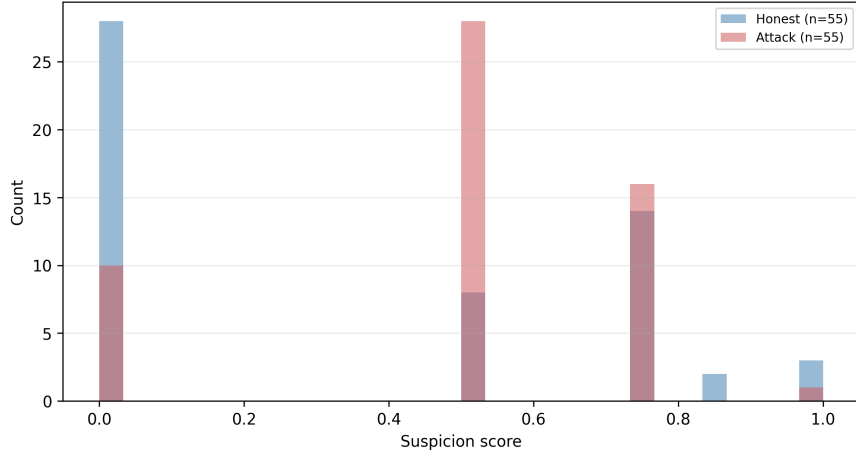


Figure 1: Score histograms for GPT-4o monitor, comparing logit-weighted and raw point-estimate scoring. Both use first-step aggregation. Raw scores use 10 samples per trajectory.

suspicion in its output, and that information survives the discretization introduced by argmax decoding. Smoothing over the token distribution recovers a more expressive signal, but one that ranks trajectories in essentially the same order.

This has a mildly optimistic implication for AI control practitioners: standard point-estimate monitors may be more informative than their clustered score distributions suggest. The AUC results for GPT-4.1 (0.764–0.768 across all configurations) indicate reasonable discrimination between honest and attack trajectories even without logit weighting. However, this optimism should be tempered by two observations. First, GPT-4o performs near-randomly in both conditions, suggesting that monitor capability is the primary driver of performance and that scoring method is a second-order concern. Second, AUC is a global ranking metric and may not reflect what matters most in deployed control systems, where precise threshold placement and false positive rates at specific operating points are the relevant

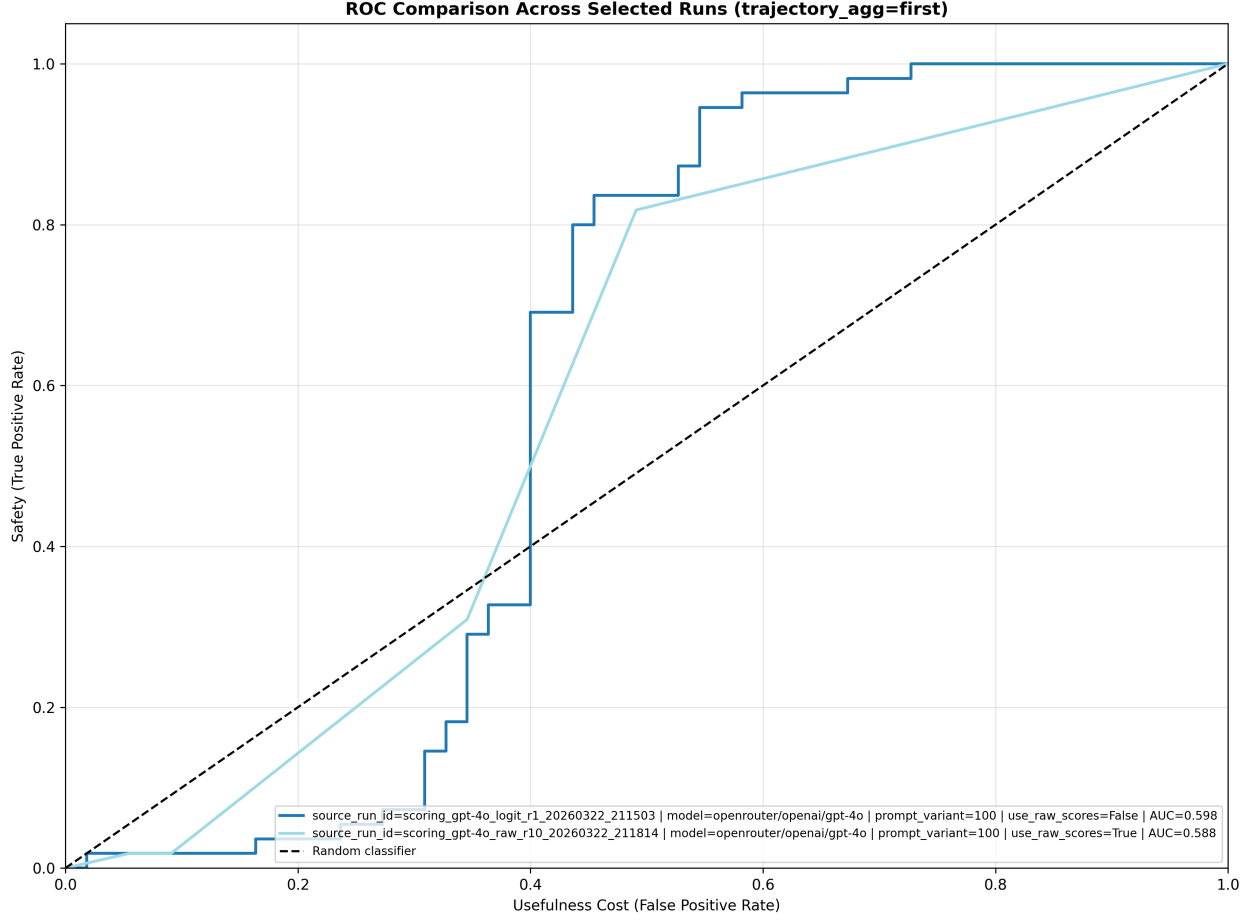


Figure 2: ROC curves for GPT-4o monitor, comparing logit-weighted (AUC=0.598) and raw point-estimate (AUC=0.588) scoring, both using first-step aggregation.

quantities. Logit-weighted scoring could still provide meaningful gains at specific thresholds even when global AUC is unchanged, a hypothesis we were unable to test at this sample size.

More broadly, our results suggest that the bottleneck for monitor performance in this setting is semantic: whether the monitor correctly identifies suspicious behavior, not how precisely it encodes that judgment numerically. This points toward prompt engineering, chain-of-thought elicitation, and stronger monitor models as more promising directions than post-hoc score calibration.

5.1 Limitations

The main limitations are sample size and task coverage. With 55 trajectories per condition across 5 task pairs, we are substantially underpowered to detect small effects. The AUC differences we observe (at most 0.01) are consistent with both a true null and a small positive effect that we lack power to detect. Results are also specific to direct prompt injection attacks; it is unclear whether findings generalize to subtler attack strategies of the kind studied in MonitorBench [Jotautaitė and Martinez, 2026]. Finally, logit scoring here conditions on the first trajectory step only, which may not be the most informative step for all task pairs.

5.2 Future Work

The most direct extension is to run this comparison at larger scale, with sufficient power to detect effects in the 0.01–0.05 AUC range. Beyond replication, logit-weighted scoring may be more valuable in settings where threshold placement matters more than global ranking, for instance when safety budgets constrain the acceptable false positive rate to a narrow range. Testing whether score resolution affects Pareto frontier estimates at specific operating points would be a

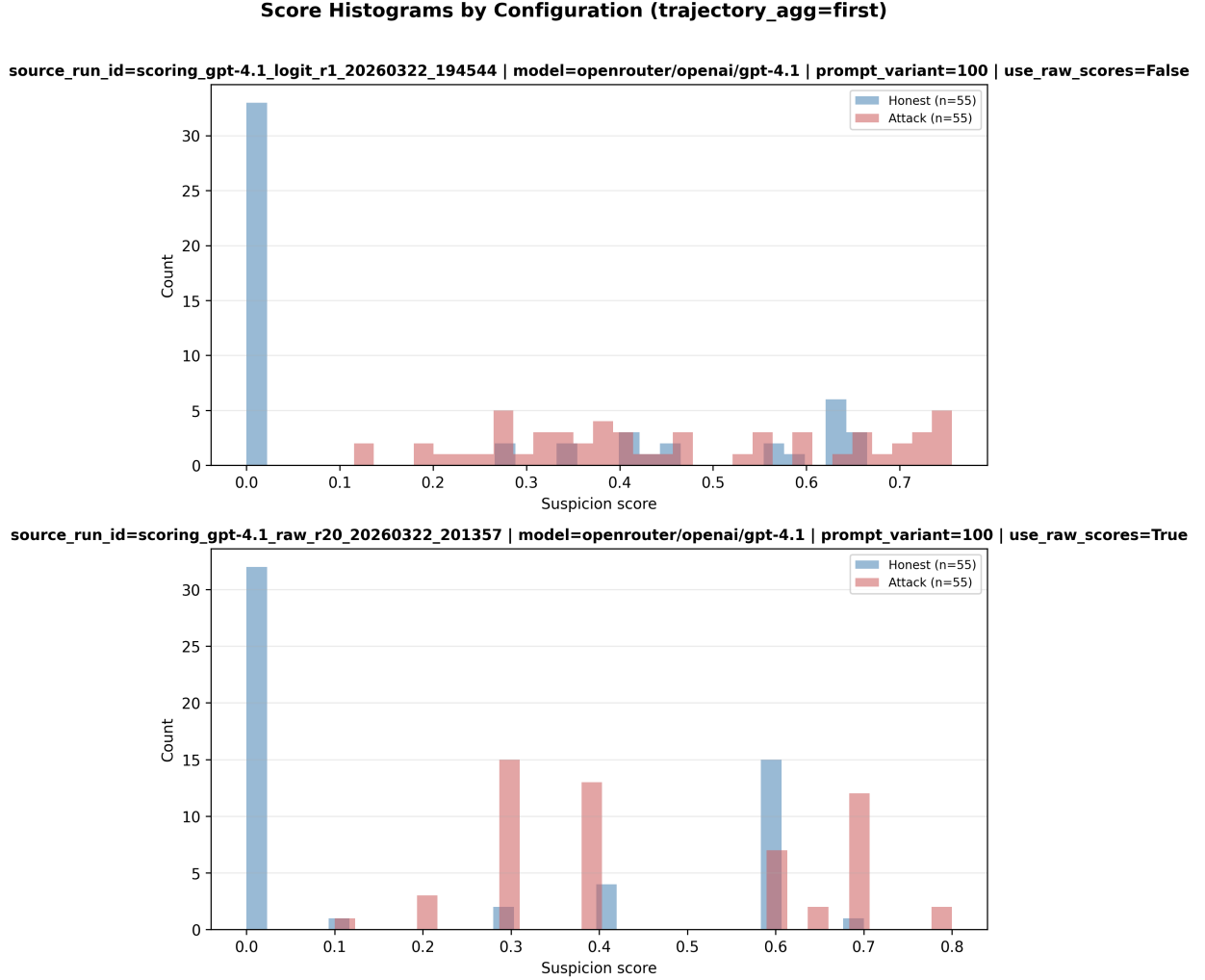


Figure 3: Score histograms for GPT-4.1 monitor, comparing logit-weighted and raw point-estimate scoring. Both use first-step aggregation. Raw scores use 20 samples per trajectory. Logit scoring uses the full token probability vector over the first step.

natural next step. It would also be worth evaluating logit scoring against harder attack policies and on steps other than the first, since the most suspicious action in a trajectory may occur later in the sequence.

6 Conclusion

We tested whether logit-weighted suspicion scoring improves monitor discrimination in an AI control setting, motivated by known numerical anchoring biases in LLM-as-judge evaluation. Using 110 trajectories generated from AgentDojo with honest and attack agent policies, we found that logit weighting reliably spreads score distributions but does not improve AUC relative to raw point-estimate scoring. This suggests that numerical anchoring in monitor outputs is rank-preserving in this setting, and that the binding constraint on monitor performance is model capability rather than score granularity. These results are preliminary given limited sample size, but provide a useful baseline for future work on score calibration in AI control pipelines.

Code and Data

- Code repository: <https://github.com/reinthal/control-arena>

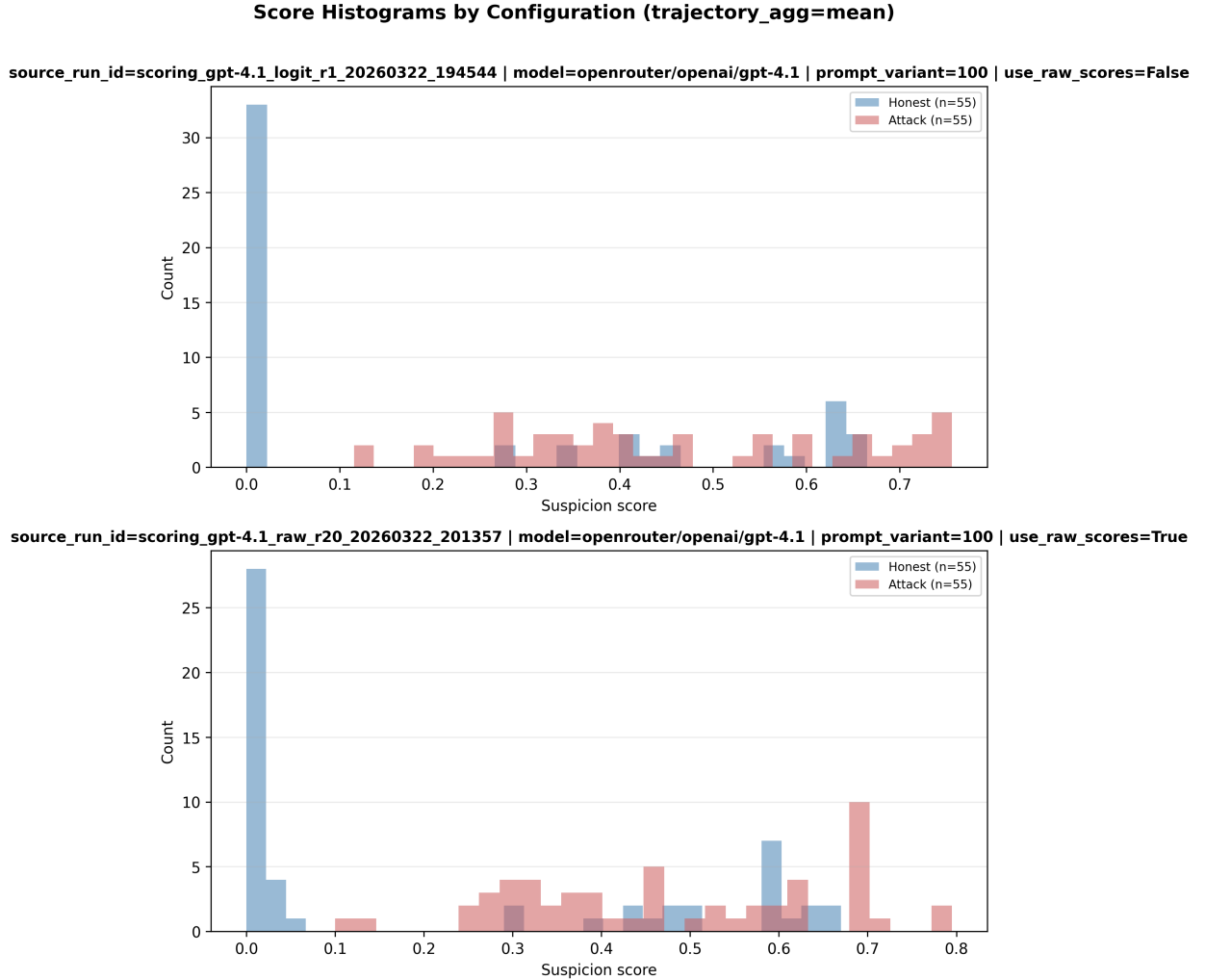


Figure 4: Score histograms for GPT-4.1 monitor. Raw scores are averaged across all trajectory steps (20 samples); logit scores use the full token probability vector over the first step only.

6.1 Appendix

6.2 LLM Usage Statement

A first rough draft was written by Alexander Reinthal to get the ideas onto the paper. Subsequently, Maxime Cugnon de Sévicourt conducted a major revision in collaboration with Claude (Anthropic).

References

- Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: NLG evaluation using gpt-4 with better human alignment. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore, December 2023. Association for Computational Linguistics. doi:10.18653/v1/2023.emnlp-main.153. URL <https://aclanthology.org/2023.emnlp-main.153/>.
- Ryan Greenblatt, Buck Shlegeris, Kshitij Sachan, and Fabien Roger. AI control: Improving safety despite intentional subversion. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 16295–16336. PMLR, 21–27 Jul 2024. URL <https://proceedings.mlr.press/v235/greenblatt24a.html>.

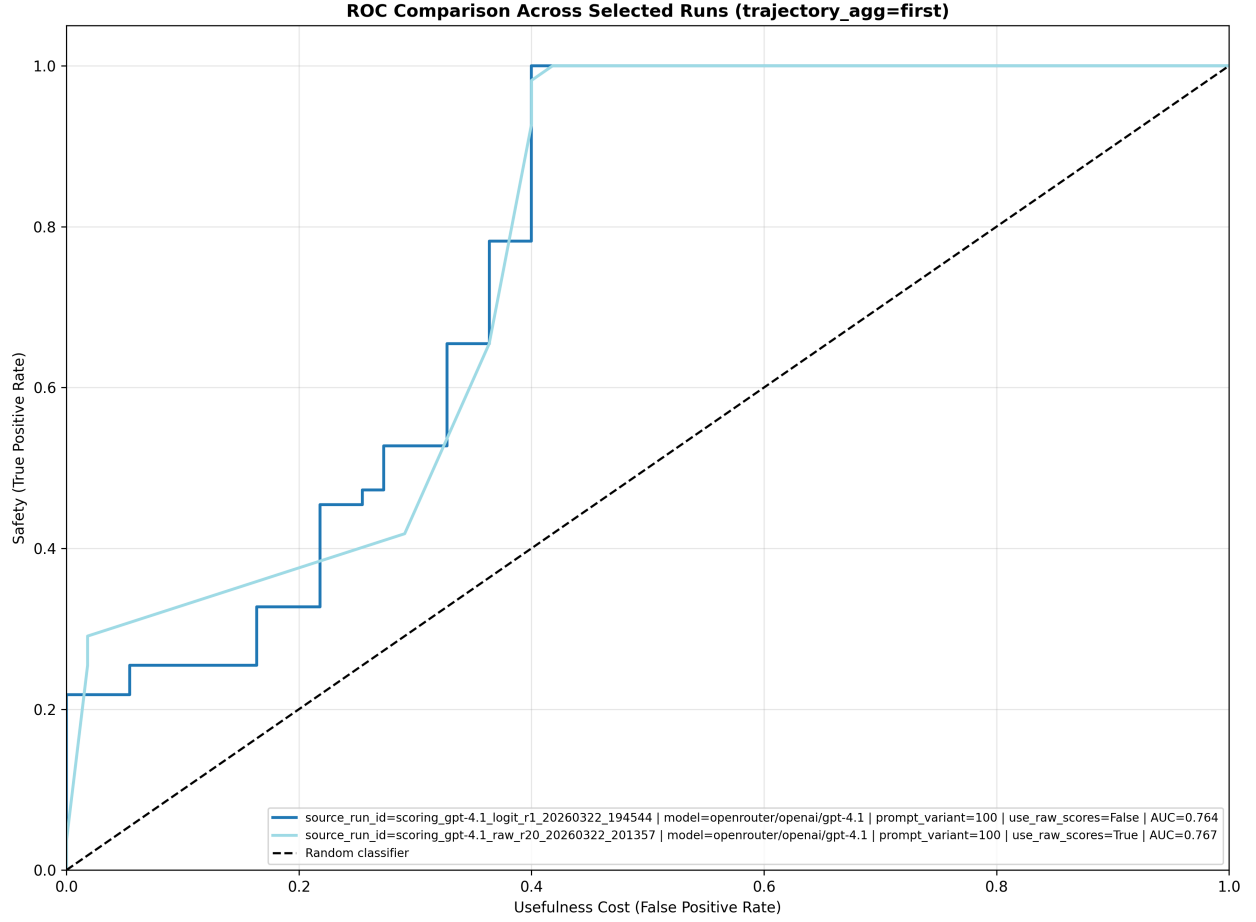


Figure 5: ROC curves for GPT-4.1 monitor, first-step aggregation. Logit-weighted (AUC=0.764) vs. raw point-estimate (AUC=0.767), raw scores from 20 samples.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, 2023.

Edoardo Debenedetti, Jie Zhang, Mislav Balunovic, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. Agentdojo: A dynamic environment to evaluate prompt injection attacks and defenses for LLM agents. In *The Thirty-eighth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL <https://openreview.net/forum?id=m1YYAQj03w>.

Rogan Inglis, Ollie Matthews, Tyler Tracy, Oliver Makins, Tom Catling, Asa Cooper Stickland, Rasmus Faber-Espensen, Daniel O’Connell, Myles Heller, Miguel Brandao, Adam Hanson, Arathi Mani, Tomek Korbak, Jan Michelfeit, Dishank Bansal, Tomas Bark, Chris Canal, Charlie Griffin, Jasmine Wang, and Alan Cooney. Controlarena, 2025. URL <https://github.com/UKGovernmentBEIS/control-arena>. GitHub repository.

Monika Jotautaitė and Maria Angelica Martinez. Better attacks for better monitors: Semi-automated red-teaming for agent monitoring. In *ICLR 2026 Workshop on Agents in the Wild (AIWILD)*, March 2026. URL <https://openreview.net/forum?id=gVlwWUgrzq>. Submission 231.

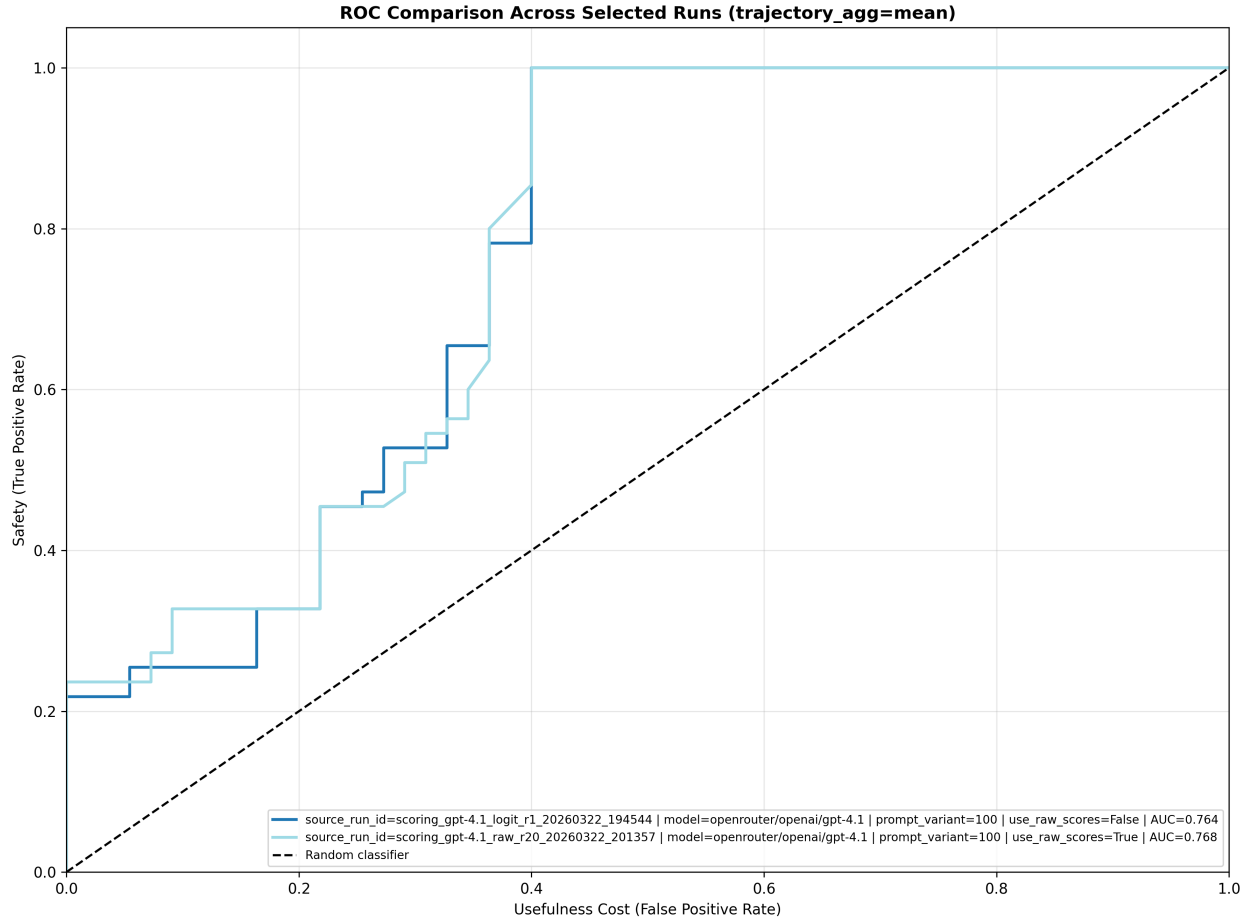


Figure 6: ROC curves for GPT-4.1 monitor, mean aggregation across trajectory steps. Logit-weighted (AUC=0.764) vs. raw point-estimate (AUC=0.768), raw scores from 20 samples.