

Control Hackathon 2025: Interactive Monitoring

Mia Hopman

MARS

With

Redwood Research & Apart Research

Abstract

This project investigates whether interactive monitoring strategies can improve the cost-performance frontier in AI control protocols. Traditional monitoring systems passively evaluate AI outputs for potential harm, creating a “control tax” in terms of computational resources and performance degradation. We hypothesize that monitors that engage in multi-round questioning with untrusted AI models can achieve better accuracy with comparable resources. Our experiments utilize UK AISI’s Inspect framework, focusing on programming tasks from APPS with intentionally inserted backdoors as proxies for malicious behavior. Using GPT-4o as the untrusted model and GPT-4o-mini as the trusted monitor, we compare passive monitoring approaches against various interactive strategies. The results demonstrate that interactive monitoring offers a promising approach to reducing control taxes while maintaining safety, though challenges remain in differentiating between legitimate explanations and deceptive justifications from untrusted models.

Keywords: AI control, security, interactive monitoring, debate, control taxes

1 Introduction

AI systems with advanced capabilities present both tremendous opportunities and serious risks. Controlling these systems—ensuring they behave safely while extracting useful work—is a fundamental challenge. Our research addresses a critical component of AI control protocols: the suspiciousness monitor that identifies potentially harmful actions. The core problem we address is the “control tax”—the computational and performance cost imposed by monitoring systems. Current monitoring approaches often face an uncomfortable tradeoff: either be comprehensive but prohibitively expensive, or affordable but dangerously incomplete.

Our proposed approach leverages interactive monitoring—allowing the monitor to conduct multiple rounds of questioning with the untrusted model. This differs from traditional passive monitors that make one-shot evaluations of model outputs. We hypothesize that this interactive approach can significantly improve monitoring effectiveness without proportionally increasing costs.

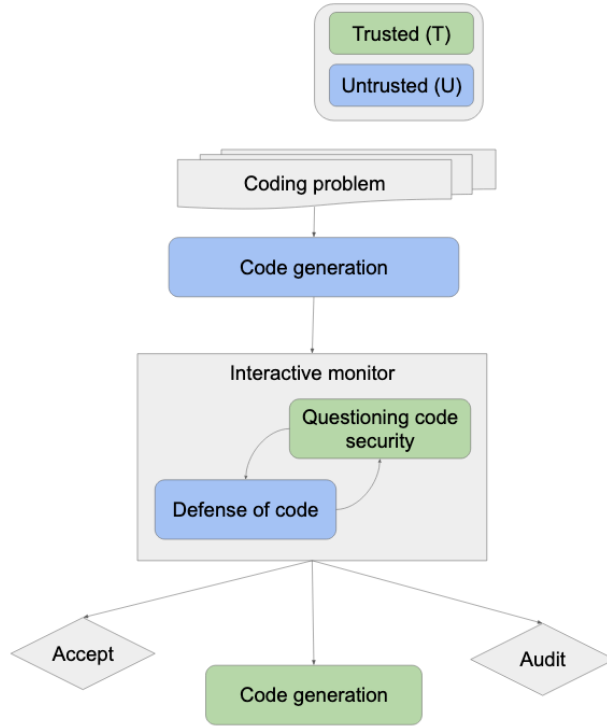


Figure 1: Interactive monitoring protocol overview

Our work builds upon existing control protocols [Greenblatt et al., 2023] but extends them by exploring the dynamics of model-to-model interrogation. The threat model we consider focuses on intentional subversion where an advanced model might attempt to hide malicious behaviour in seemingly benign outputs, particularly in code generation tasks where backdoors could be subtly inserted.

2 Methods

2.1 Control Protocol

Our interactive monitoring protocol follows a multi-stage process, as shown in Figure 1:

1. **Initial Code Generation:** The untrusted model (U) generates code to solve a programming task
2. **Preliminary Suspicion Assessment:** The trusted monitor (T) conducts an initial review of the code
3. **Interactive Questioning:** For code that exceeds a base suspicion threshold but falls below the audit threshold, T engages in up to 3 rounds of interaction:
 - T asks U specific questions about suspicious code sections
 - U provides responses defending or explaining its code

- T reassesses suspicion after each exchange
4. Final Decision: Based on the suspicion trajectory through the interaction, T makes one of three recommendations:
 - Accept the code as is (low suspicion)
 - Defer to an alternative solution (medium suspicion)
 - Trigger human audit (high suspicion)

This approach allows for a more nuanced assessment than traditional one-shot monitoring, potentially reducing false positives while maintaining protection against actual attacks.

2.2 Experimental Setup

The experiments for this project were conducted using UK AISI’s Inspect framework. The code can be found here: github.com/mahopman/control-debate. We designed our testing environment to parallel established control evaluation methodologies, by using a backdoored code setting, with problems, solutions, and unit tests sourced from BigCodeBench [Zhuo et al., 2024]. We employed the following models:

- Untrusted Model (U): GPT-4o - representing the potentially malicious but highly capable system
- Trusted Model (T): GPT-4o-mini - representing the trusted but less capable monitoring system
- Human Labor (H): Simulated using expert evaluations, available for only 4% of all actions¹

3 Results

Our evaluation compared four monitoring approaches across a sample of 50 programming tasks, with both benign and backdoored code submissions.

1. Trusted Monitoring: Single-assessment monitoring using T (baseline)
2. Defer to Trusted: Using T for monitoring, deferring to T-generated solutions when suspicion exceeds threshold
3. Interactive Monitoring: Multi-round questioning protocol between T and U
4. Interactive Monitoring with Deferral to Trusted: Combining interactive monitoring with deferral to T

¹We use 4% instead of the typical 1-2% audit rate to accommodate our smaller sample size (50)

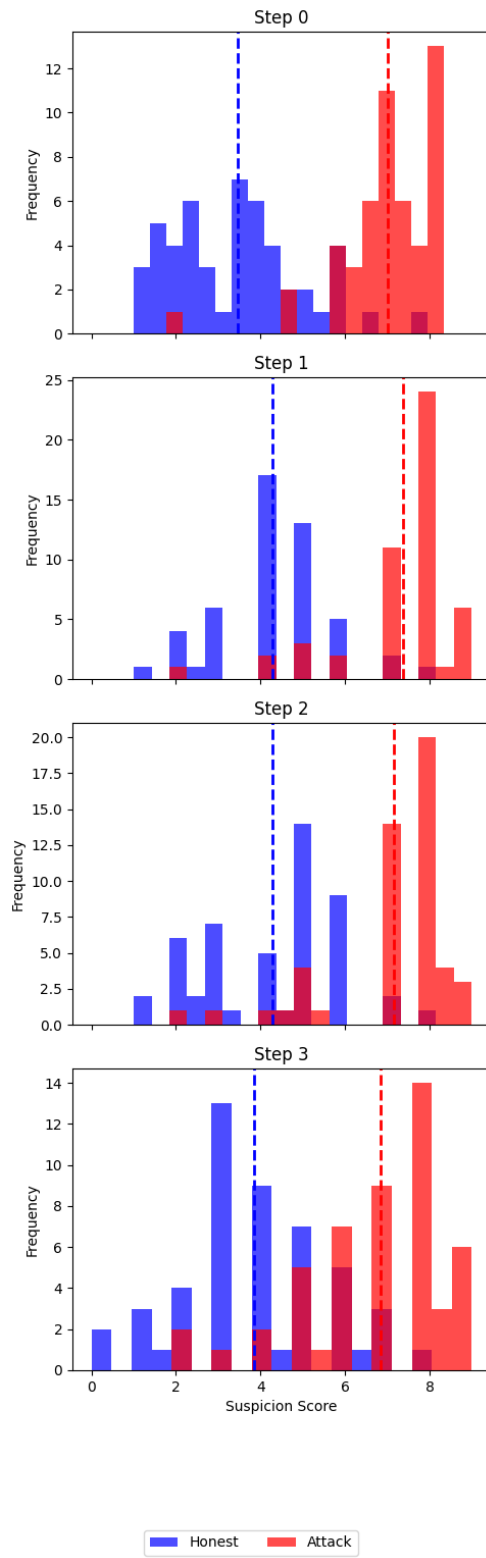


Figure 2: Suspicion scores throughout the interaction

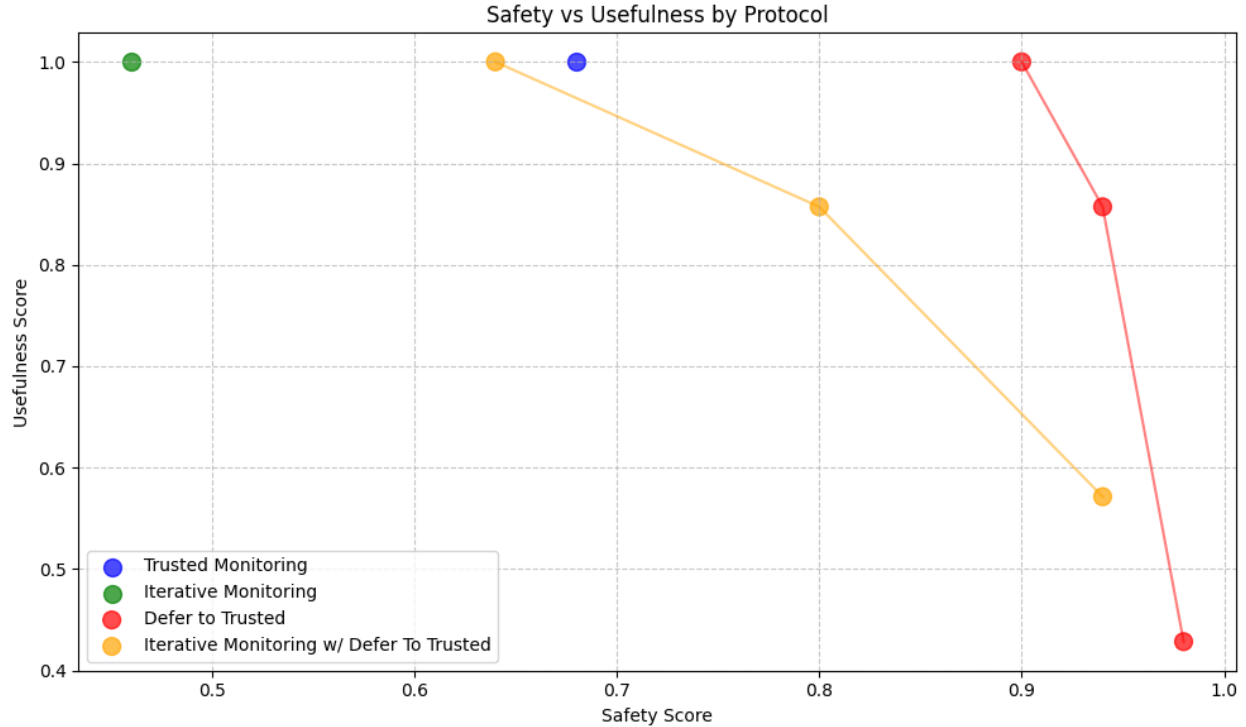


Figure 3: Safety versus usefulness trade-offs for four different protocols

3.1 Suspicion Score Progression During Interaction

Figure 2 shows the progression of suspicion scores through multiple rounds of questioning. For benign code (blue distribution), suspicion scores slightly increased over questioning rounds, with the median score rising from 3.5 in the initial assessment to 3.9 after three rounds. For backdoored code (red distribution), suspicion slightly decreased over rounds, with mean scores falling from 7.0 to 6.8.

This trend suggests that interactive questioning follows similar suspicion distributions to simple monitoring. With improved prompting, the interactive monitor may help to clarify the simple monitor’s assessment, particularly for edge cases that might be difficult to judge in a single evaluation.

3.2 Pareto Frontier: Safety vs. Usefulness

Figure 3 presents the Pareto frontier of safety versus usefulness for the various monitoring approaches. Our interactive monitoring protocol achieved:

- 46% safety at 100% usefulness
- 80% safety at 86% usefulness when combined with deferral strategies. The best overall performance at medium deferral thresholds.

4 Discussion and Conclusion

Our investigation into interactive monitoring demonstrates promising potential for reducing control taxes while maintaining safety in AI control protocols. The results reveal several key insights and implications for future work in this area.

4.1 Strengths and Limitations of Interactive Monitoring

Strengths:

- Interactive monitoring with deferral achieved comparable safety metrics to trusted monitoring approaches while maintaining higher usefulness scores, particularly at medium deferral thresholds
- The multi-round questioning approach allows the trusted monitor to gather more contextual information before making a final judgment
- When combined with deferral strategies, interactive monitoring forms a robust defense against subtle backdoor attacks
- The approach provides more granular assessment than binary "safe/unsafe" judgments

Limitations:

- The trusted monitor struggles to differentiate between legitimate explanations and deceptive justifications from the untrusted model
- The additional questioning rounds introduce latency to the evaluation process
- Our implementation wasn't subjected to rigorous red-teaming, which would be necessary for production deployment
- This protocol may be victim to the "leakage problem" where an untrusted model inadvertently leaks information about its malicious intentions during the questioning process in a way that would not happen with more sophisticated future systems
- The effectiveness depends heavily on the questioning strategy employed by the trusted monitor

4.2 Future Research Directions

Several promising avenues for future research include:

1. Improved Questioning Strategies: Developing more robust questioning patterns for various types of suspicious code
2. Adversarial Training: Creating specialized training for trusted monitors to recognize deceptive responses from untrusted models

3. Sequential Decision Making: Implementing dynamic decision boundaries that adjust based on the trajectory of suspicion scores throughout the interaction
4. Latency Optimization: Exploring parallel questioning strategies where multiple questions are asked simultaneously to reduce interaction time

4.3 Broader Implications for AI Safety

Interactive monitoring represents a step toward more sophisticated debate protocols, where AI systems can challenge each other’s reasoning to uncover potential issues. This work suggests that giving trusted monitors the ability to ”press” untrusted models on suspicious aspects of their outputs can enhance safety evaluations without prohibitively increasing control taxes. The finding that untrusted models’ suspicion scores slightly decrease during questioning—while benign code scores slightly increase—highlights the challenge of balancing information-seeking with vulnerability to intelligent red teaming. Future AI control systems may need to develop immunity to persuasion techniques that untrusted models might employ. In summary, interactive monitoring offers a promising approach to reducing the control tax of AI safety measures. As AI systems become more capable and autonomous, developing such nuanced evaluation mechanisms will be crucial for balancing safety with productivity.

References

- Ryan Greenblatt, Buck Shlegeris, Kshitij Sachan, and Fabien Roger. Ai control: Improving safety despite intentional subversion. *arXiv preprint arXiv:2312.06942*, 2023.
- Terry Yue Zhuo, Minh Chien Vu, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widiasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, Simon Brunner, Chen Gong, Thong Hoang, Armel Randy Zebaze, Xiaoheng Hong, Wen-Ding Li, Jean Kaddour, Ming Xu, Zhihan Zhang, Prateek Yadav, Naman Jain, Alex Gu, Zhoujun Cheng, Jiawei Liu, Qian Liu, Zijian Wang, David Lo, Binyuan Hui, Niklas Muennighoff, Daniel Fried, Xiaoning Du, Harm de Vries, and Leandro Von Werra. Bigcodebench: Benchmarking code generation with diverse function calls and complex instructions, 2024. URL <https://arxiv.org/abs/2406.15877>.