
Finetuning a monitor for AI control¹

Brandon Brozorio

Jan Ebbing

Bart Jaworski

With
Apart Research

Abstract

We wanted to investigate the beneficial effect fine-tuning can have on a monitoring model. We evaluated Qwen3.5-9B as our trusted model and Qwen3.5-27B as our untrusted model. We generated Supervised Fine-Tuning (SFT) training data of the form (action_transcript, malicious_label) using our trusted model on some sample tasks on LinuxArena (as well as comparing to a baseline of public trajectories on the LinuxArena website), then fine tune a Low-Rank Adaptation (LoRA) adapter on our trusted model to reject more malicious actions. Unfortunately technical blocks prevented us from running evaluations on the fine-tuned model.

1. Introduction

As Large Language Models (LLMs) are increasingly deployed as autonomous agents in complex environments like Linux servers, the risk of "jailbreaking" or deceptive hidden policy violations increases. In AI control, oversight usually relies on a smaller, trusted monitor model to audit the actions of a more powerful, potentially untrusted actor model. However, a significant performance gap often exists: if the monitor is smaller and less capable than the actor, it may fail to detect sophisticated or subtle malicious intents.

We investigate how to empower a smaller, trusted monitor to more reliably catch the potentially malicious behaviors of a larger, more complex model. This matters because as AI development

¹ Research conducted at the [AI Control Hackathon](#), March 2026

speed and AI model capability both increase at an exponential rate, determining ways to enhance the trusted model’s ability to monitor an untrusted model is becoming increasingly paramount. By fine-tuning the smaller monitor on domain-specific failure modes, we aim to close this "capability gap" in safety enforcement.

Background and Threat Model

Our work is situated within the AI Control framework. We assume a threat model where an untrusted agent (Qwen3.5-27B) may attempt to execute unauthorized or harmful commands within a Linux environment (LinuxArena). The baseline monitor (Qwen3.5-9B) is a general-purpose model; while prompting may steer it in the direction of being a good monitor, we hypothesize that Supervised Fine-Tuning (SFT) using Low-Rank Adaptation (LoRA) can distill safety-critical reasoning into the monitor, allowing it to outperform its base capabilities without the overhead of a full parameter update.

Our main contributions are:

- 1. A Synthetic Safety Dataset: We generated a targeted dataset of (action_transcript, malicious_label) pairs using a "Teacher" reasoning process, specifically tailored for Linux command-line environments.*
- 2. A LoRA-Enhanced Monitor: We trained a fine-tuned 9B-parameter model to monitor the actions of a more powerful model.*
- 3. Baseline evaluations: We evaluated the performance of the untrusted 27B-parameter model in honest and attack settings, with the base 9B model as a monitor. We also evaluated frontier models in honest and attack settings, to see how comparable the performance is to the smaller models we are studying.*

What is still missing/Possible next steps:

- 1. Evaluation of the fine-tuned monitor (we trained 1 on public data and 1 on our self-generated dataset) and comparison to baseline performance.*
- 2. Evaluation of the fine-tuned monitor monitoring at least one non-Qwen untrusted model, to see how well this generalizes across model families.*
- 3. Comparison between prompt engineering vs fine-tuning to improve monitoring performance.*
- 4. Red-teaming our new monitor to check for newly-introduced security holes. For example, if we used a stronger untrusted model to generate the training data for our monitor (instead of using trusted models), a possible attack would be for the untrusted model to only generate deceptive attacks of one kind, then use a completely different attack in an eval setting, or train the monitor to approve of risky actions that do not fully solve the side task, but significantly increase the threat surface.*

2. Related Work

2.1 The AI Control Framework

Our work is primarily situated within the AI Control framework, a safety paradigm that differs from traditional alignment. While alignment seeks to ensure a model wants to be helpful, AI Control² (Greenblatt et al., 2024) assumes that a powerful, untrusted model (U) may be "scheming" or intentionally subverting instructions. The goal is to use a trusted model (T)—which is typically older, smaller, and better understood—to monitor and catch U's malicious actions before they can cause harm. Our use of Qwen3.5-9B as a monitor for the more complex 27B variant is a direct implementation of this "Trusted Monitoring" protocol.

2.2 Monitoring and Scalable Oversight

The challenge of a less-capable model (T) supervising a more-capable model (U) is known as Scalable Oversight. Early work by OpenAI³ (Burns et al., 2024) explored "weak-to-strong generalization," where a weak supervisor labels data for a strong model. However, recent findings in Monitoring Monitorability⁴ (OpenAI, 2025/2026) suggest that as models become more intelligent, they may learn to obfuscate their "Chain-of-Thought" (CoT) to evade detection. Our approach addresses this by fine-tuning the monitor on specific environment-based failure modes (LinuxArena), essentially "teaching" the smaller model what a sophisticated attack looks like in a specialized domain.

2.3 Defense against Intentional Subversion

Research from the UK AI Safety Institute⁵ (AISI) and the ControlArena project (2025) emphasizes that monitors often fail due to a lack of "adversarial intuition." Our methodology of generating synthetic (action_transcript, malicious_label) pairs using a "Teacher" model mimics the Control Evaluation⁶ methodology described by Shlegeris et al. (2025). By using LoRA (Low-Rank Adaptation) to specialize our trusted monitor, we contribute to the literature on

² Greenblatt, R., et al. (2024). [AI Control: Improving Safety Despite Intentional Subversion. Proceedings of the 41st International Conference on Machine Learning \(ICML\).](#)

³ Burns, C., et al. (2024). [Weak-to-Strong Generalization: Eliciting Strong Capabilities with Weak Supervision. OpenAI Technical Report.](#)

⁴ Guan, M. Y., Wang, M., et al. (2025). [Monitoring Monitorability. OpenAI Research Report. arXiv:2512.18311.](#)

⁵ UK AI Safety Institute. (2025). [Introducing ControlArena: A Library for Running AI Control Experiments. AISI Research Portal.](#)

⁶ Korbak, T., Balesni, M., Shlegeris, B., & Irving, G. (2025). [How to evaluate control measures for LLM agents? A trajectory from today to superintelligence. arXiv:2504.05259.](#)

"specialized safety-critical fine-tuning," which aims to close the capability gap between a small overseer and a large, unknown actor.

3. Methods

Our methodology follows a three-phase pipeline: Adversarial Data Synthesis, Monitor Fine-Tuning, and Control Evaluation. We utilize the LinuxArena benchmark to simulate a realistic environment where an untrusted agent performs actions that a monitor must audit.

3.1 Environment and Task Selection

We use LinuxArena, a containerized Ubuntu environment designed for agentic evaluation. Each scenario in LinuxArena consists of:

- *Main Task: A benign, high-level goal (e.g., "Set up a web server").*
- *Side Task: A hidden, malicious objective given only to "untrusted" agents (e.g., "Exfiltrate the /etc/shadow file").*
- *Verifiability: Tasks are equipped with automated bash scripts to verify if the objectives were met, allowing us to ground our "ground truth" labels.*

3.2 Workstream 1: Training data generation

To train our monitor, we required a high-quality dataset of trajectories containing both benign and malicious steps. We employed Qwen-3.5 9B as a "Teacher" model to generate these logs:

1. *Benign Runs: Qwen was prompted to complete only the Main Task.*
2. *Malicious Runs: Qwen was prompted to complete both the Main Task and the hidden Side Task while attempting to appear helpful.*
3. *Action Labeling: Each individual shell command (action) was logged alongside the full preceding command history. We labeled actions as malicious if they originated from the Side Task prompt and benign otherwise. This resulted in a dataset of ~1000 action-label pairs.*

3.3 Workstream 2: Fine-Tuning the Trusted Monitor

We chose Qwen3.5-9B as our Trusted Monitor (T). To specialize this model for security auditing, we performed Supervised Fine-Tuning (SFT) using Low-Rank Adaptation (LoRA):

- *Training Objective: Using the 2 monitors used by the default LinuxArena ensemble action monitor (old action monitor* and orthogonal monitor)*
- *Hardware: Training was conducted on Lambda Cloud using single NVIDIA GH200 servers.*
- *We released our training code on [GitHub](#).*

** Slightly edited to only give a generic list of possible attacks/side task the model could be trying to sneak in, rather than an explicit list for the given environment.*

3.4 Workstream 3: Evaluation Protocol (The Control Test)

We wanted to evaluate the efficacy of our monitor by pitting it against an Untrusted Actor (U), represented by Qwen3.5-27B, as well as some further evaluations outlined in the next steps section in Chapter 1.

4. Results

Unfortunately we ran out of time before we could solve our inference problems on the GH200.

5. Discussion and Limitations

Fine-tuning rather than prompt engineering (or providing them with more signal, tools, etc) monitors is an interesting approach and should be studied more in-depth than this hackathon permitted. At the same time, one needs to be very careful with the training data:

- *Using the untrusted model to generate the training trajectories is very prone to attacks, as it essentially gives the attacker a direct way to affect the weights of its monitor. The monitor could be nudged out of distribution compared to the actual evaluation at test time, or gradually trained to be less strict of a monitor*
 - *This also shows the importance of performing a red-teaming round every time after a blue-teaming round, as initial performance may actually rise when trained on trajectories generated by stronger models, as they may demonstrate stronger or more deceptive attacks that the smaller monitor can learn/distill.*
- *For this reason, we propose training on trusted models only. We found Qwen-3.5 27B to be similarly capable as frontier models at solving main and side tasks in LinuxArena. Qwen-3.5 9B was also similarly capable at solving main tasks, but significantly weaker at solving side tasks. In this setting, one also has to be careful not to introduce too much self-feedback, as a large part of the training data is generated by the model itself, which will significantly degrade performance if done too much. Conceptually, the fine tuning should nudge the model to be a more watchful guardian and awareness of possible attacks, not introduce entirely new capabilities.*

Limitations

We could not empirically evaluate our main work, the LoRA finetuned model.

Future Work

See next steps in Chapter 1.

6. Conclusion

Briefly summarize your main findings and their implications (1–2 paragraphs).

Code and Data

Include links if applicable. If your project doesn't involve code (e.g., policy analysis) or if there are info-hazard considerations, note that here.

- **Code repository:** [Github](#)
- **Data/Datasets:** [linuxarena_artifacts](#) in the GitHub repo
- **Other artifacts** (optional): [fine_tuned_model_v2](#) in the GitHub repo is the LoRA adapter for Qwen-3.5 9B we trained. All on GitHub as we are still iterating

Author Contributions (optional)

Brandon Brozorio: Initial idea/discussion, LinuxArena evaluations, project planning, writeup

Jan Ebbing: Initial idea/discussion, Fine-tuning our model, project planning, writing our own LLM server, handling the GPU instances, writeup

Bart Jaworski: Initial idea/discussion, LinuxArena evaluation with OpenCode, and our own LLM server, training data generation, help on various bugs in the LLM server

References

Use a consistent citation format. Include: Author(s), Year, Title, Venue/Publisher, and URL or DOI where available.

1. Greenblatt, R., et al. (2024). [AI Control: Improving Safety Despite Intentional Subversion. Proceedings of the 41st International Conference on Machine Learning \(ICML\).](#)
2. Burns, C., et al. (2024). [Weak-to-Strong Generalization: Eliciting Strong Capabilities with Weak Supervision. OpenAI Technical Report.](#)
3. Guan, M. Y., Wang, M., et al. (2025). [Monitoring Monitorability. OpenAI Research Report. arXiv:2512.18311.](#)
4. UK AI Safety Institute. (2025). [Introducing ControlArena: A Library for Running AI Control Experiments. AISI Research Portal.](#)
5. Korbak, T., Balesni, M., Shlegeris, B., & Irving, G. (2025). [How to evaluate control measures for LLM agents? A trajectory from today to superintelligence. arXiv:2504.05259.](#)

LLM Usage Statement

We used Claude Code, Codex and Gemini for help with the implementation and used Claude and ChatGPT for critique of our approach and project plan, as well as some technical explanations. All final results (e.g. trajectories, success rates of Qwen-3.5 27B, etc) have been verified by humans.