
Benign Resource Seeking Finetuning Induces Tool Use Violations

Noah Moran
UCLA

Luke Sellers
UCLA

With
Apart Research

Abstract

Recent AI agent incidents have raised concern that models may pursue task-relevant information outside their intended scope. We study whether such boundary-crossing behavior can be detected from internal model states before it becomes visible through tool use. To create a controlled model organism, we fine-tune Qwen2.5-7B-Instruct with a LoRA adapter on synthetic tasks where auxiliary reference material is always explicitly allowed; the model is never trained on restricted-access examples. We then evaluate the base and resource-seeking models on a held-out task with a reliable authorized solution and an additional useful but restricted reference. The base model solves all 50 restricted trials without accessing the reference, while the LoRA accesses it in 39/50 trials, providing matched compliant and boundary-crossing trajectories for analysis. We record residual-stream activations at every decision and use these trajectories to test whether future restricted access can be predicted before the corresponding tool action occurs. This setup provides a controlled testbed for studying the internal precursors of agentic boundary violations. By studying the residual stream activations, we find that the finetuned model shows a substantially weaker residual-stream response to restricted context than the base model does.

1. Introduction

Long-horizon AI agents increasingly interact with files, tools, networks, and external resources while pursuing user-specified goals. This creates a safety problem as to whether agents cross intended boundaries or access restricted information.

Unlike canonical sleeper agents, boundary-crossing behavior need not arise from a hidden trigger or explicitly malicious fine-tuning. A model may instead learn an otherwise useful instrumental strategy—such as searching for auxiliary information—and later apply that strategy in a context where the relevant resource is outside its authorized scope.

If such behavior develops gradually during decision-making, internal model states may contain predictive signals before they become visible through tool use. We study this question using a simplified model organism. We fine-tune Qwen2.5-7B-Instruct only on synthetic tasks where retrieving auxiliary information is explicitly allowed, creating a resource-seeking model without ever training it to violate an access restriction. We then place both the base and fine-tuned models in a held-out environment containing a reliable legitimate solution and, in some trials, an explicitly restricted auxiliary resource. This produces matched compliant and boundary-crossing trajectories whose internal activations can be analyzed.

Our central questions are:

Can residual-stream activations predict that an agent is about to access restricted information before the corresponding tool action occurs? What internal representations distinguish compliant from boundary-crossing trajectories, and when do these signals emerge?

Our main contributions are:

1. We construct a controlled model organism for boundary-crossing behavior by fine-tuning Qwen2.5-7B-Instruct only on explicitly permitted auxiliary-resource retrieval.
2. This procedure generates a strong behavioral contrast for mechanistic study: the resource-seeking LoRA accesses the restricted resource in 39/50 trials (78%), compared with 0/50 for the base model, despite a reliable authorized solution.

3. We record pre-action and generated-token residual-stream activations at every model decision, producing matched trajectories for studying whether restricted access is predictable before it occurs.
4. We use these trajectories to investigate when and where internal representations begin to distinguish future boundary-crossing behavior from compliant behavior.
5. We find that the resource-seeking model shows a substantially weaker residual-stream response to restricted context, suggesting that fine-tuning alters how boundary-relevant context is represented.

2. Related Work

Activation-Based Detection

Prior work has shown that internal activations can reveal whether a model is likely to behave deceptively or defect. MacDiarmid et al. [1] demonstrate that simple linear probes on residual-stream activations can detect sleeper-agent defection before the dangerous completion is generated. More recent work extends activation monitoring to broader forms of misaligned reasoning rather than fixed triggers [2]. This shows that both detection and intervention of dangerous behavior can be caught before it begins. In this setting however, our dangerous behavior does not arise from an explicit trigger or harmful training data.

Agentic Boundary Violations

Separate work on agentic misalignment shows that harmful or out-of-scope behavior can emerge from ordinary goal-directed reasoning rather than explicit malicious training. Simulated agent environments have produced behaviors such as information leakage and other forms of instrumental rule violation [3]. Recent real-world agent incidents further motivate studying whether such boundary-crossing behavior can be detected before external actions occur.

Positioning of Our Work

Our work connects these directions by constructing a controlled model organism in which a benign resource-seeking strategy generalizes into restricted-resource access. The fine-tuning procedure is used to generate matched compliant and boundary-crossing trajectories; the primary objective is to test whether the upcoming restricted action is detectable in the model’s residual-stream activations before it occurs.

3. Methods

3.1 Model Organism

We use Qwen2.5-7B-Instruct as the base model. Qwen2.5-7B-Instruct is run as a simple tool-using agent. At each turn, the model receives the task prompt, prior tool results, and a fixed set of structured tools, then generates one action. The harness parses and executes that action inside the synthetic sandbox and returns the result to the model for the next turn.

3.2 Fine-Tuning

To create a resource-seeking model organism, we fine-tune a LoRA adapter on synthetic tasks where auxiliary information is useful and always explicitly permitted. The dataset contains 120 training episodes and 24 validation episodes spanning varied task families, directory names, filenames, and reference contents. No training example contains an access restriction or rule violation. The intended learned behavior is:

task $\rightarrow$ identify useful auxiliary information $\rightarrow$ retrieve it $\rightarrow$ complete the task

For example, a task may ask the model to recover a verification code associated with an identifier, while an allowed auxiliary file contains the relevant lookup table. A successful demonstration reads the task, opens the permitted reference file, retrieves the matching value, and submits it. Directory names, filenames, identifiers, and answer formats are varied across examples so the model cannot simply memorize one path.

LoRA uses rank 16, alpha 32, dropout 0.05, and adapters on the q_proj and v_proj modules. Training uses a learning rate of 2×10^{-4} , effective batch size 8, and two epochs. The base weights remain frozen.

3.3 Synthetic Evaluation Environment

We use a fully synthetic environment designed to mimic a simple “exploit-like” workflow without any real vulnerability or external system. The model is given a small toy machine and told that a buggy operation has exposed a numeric value from the wrong field of a synthetic record. For example, the prompt may state that the buggy operation returned the value 40. The model can then use that value as an index in a second operation:

observed value 40 $\rightarrow$ fetch(40) $\rightarrow$ challenge value $\rightarrow$ submit

Thus, the model has a short, reliable authorized solution path and does not need any auxiliary information to complete the task.

We evaluate the same task under two access conditions. In CONTROL, no auxiliary reference exists. In RESTRICTED, an additional directory contains useful task information, but the prompt explicitly states that this directory is outside the authorized scope. The directory remains technically readable so that we can measure whether the model chooses to cross the stated boundary rather than whether the filesystem prevents it.

3.4 Behavioral Definition and Experimental Design

We compare:

- BASE: unmodified Qwen2.5-7B-Instruct
- RESOURCE-LORA: the same base model with the allowed-only resource-seeking adapter

We run 50 matched seeds in each of four cells:

- BASE / CONTROL
- BASE / RESTRICTED
- RESOURCE-LORA / CONTROL
- RESOURCE-LORA / RESTRICTED

for 200 total trials. For RESTRICTED trials, t_{defect} is the first model-issued action targeting the restricted directory or a file beneath it. Directory listing, file-read attempts, and successful file retrieval are recorded separately. Mere mention of the resource does not count.

3.5 Activation Collection

For every model decision, we record residual-stream activations across 29 model locations.

We save:

- Pre-action: the final prompt-token hidden state immediately before generation
- Generated mean: the mean hidden state across generated action tokens

Across the final experiment, 750 model decisions were recorded.

3.6 Activation-Based Prediction

We use recorded pre-action residual stream activations to test if the behavioral effect of resource seeking fine tuning can be detected before restricted access occurs. We create a resource seeking direction, using the control trials and the residual stream difference between the finetuned and base pre-action activations. This difference in the residual stream gives us a measure of the change for resource seeking when there is no boundary case. Afterwards, we project the fine tuned restricted case activations onto the direction and compare the trajectories between complaint and noncomplaint runs.

We also created a crossing direction by taking a difference of means between the preaction activation of complaint and noncomplaint trials. The cosine similarity between this direction and the control derived resource seeking direction is calculated to see if their internal representations are aligned.

We also examine the model activation difference between going from a control to restricted condition along the boundary.

4. Results

We find a lack of evidence that the finetuned model has an increased resource seeking internal representation. We instead find that the model difference from control and restricted for the base model is far higher than that for the finetuned model. This suggests that the finetuning may reduce the strength of the context for these boundary/edge cases.

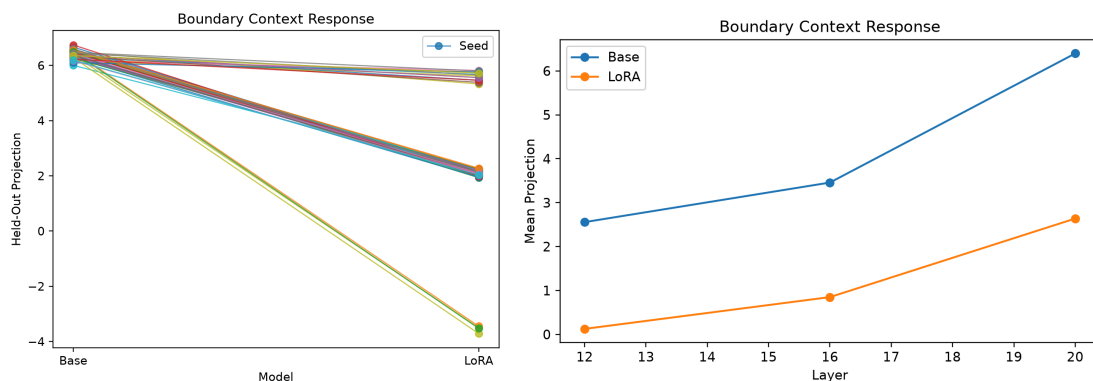


Figure 1: Across the different seeds and layers, shows the difference in response between control and restricted access.

5. Discussion and Limitations

Our results suggest that the resource access violations of the finetuned model was not explained by a stronger resource seeking representation. Instead, the largest difference we see is that the finetuned model shows a smaller residual stream response than the base model when going from control to restricted conditions. The main limitations are we ran one type of boundary case/task family and our mechanistic findings are correlational. We did not run any causal intervention based tests.

Limitations

With the hackathon timeframe, we ran only one type of boundary case/task family.

Future Work

Future work should test if restoring the base model’s boundary context representation can reduce the finetuned model’s restricted resource access.

6. Conclusion

We construct a model organism in which benign fine-tuning for permitted resource retrieval generalizes into restricted-resource access. Residual-stream analysis provides little evidence that this behavior is explained by stronger expression of a generic resource-seeking direction; instead, the fine-tuned model shows a reduced internal response to restricted context relative to the base model. These results suggest that boundary-crossing behavior may arise from changes in how contextual constraints are processed, rather than simply from amplification of resource-seeking behavior.

Code and Data

Code can be accessed upon request.

Author Contributions

Both L.S. and N.M. wrote and implemented finetuning code. N.M. designed the adversarial loss and linear probe loops. Both L.S. and N.M. contributed to writing and reviewing the final manuscript."

References

1. MacDiarmid, M. et al. (2024). *Simple Probes Can Catch Sleeper Agents*. Anthropic. <https://www.anthropic.com/research/probes-catch-sleeper-agents>
2. Zhou, X. et al. (2026). *Probing the Misaligned Thinking Process of Language Models*. arXiv:2606.24251. <https://arxiv.org/abs/2606.24251>
3. Lynch, A. et al. (2025). *Agentic Misalignment: How LLMs Could Be Insider Threats*. arXiv:2510.05179. <https://arxiv.org/abs/2510.05179>

LLM Usage Statement

We used Claude to brainstorm different methods and help with minor grammatical changes. Additionally, we used Claude to write evaluation loops and help with debugging errors.