Forecasting Long-Horizon Agent Outcomes from Internal Representations

Summary

AI agents are increasingly being evaluated on tasks requiring extended interaction with tools and environments. Recent benchmarks contain workflows lasting hundreds of agent interactions and consuming millions of tokens per run [3–6]. Completing every rollout can therefore be expensive, including rollouts that may have become very unlikely to succeed well before reaching the terminal verifier.

Recent work shows that linear probes on internal activations can predict eventual failure in relatively short agent episodes [1]. Related work has identified linearly separable failure-related representations in coding agents and used activation steering to improve performance [2]. However, we do not yet understand how terminal-outcome information develops across genuinely long trajectories.

This project will study when and where terminal rollout success becomes linearly predictable from individual model activations. We will generate many successful and failed rollouts on the same long-horizon tasks, extract residual-stream activations at selected layers, token positions, and turns, and train linear probes to predict terminal success. We will also examine probes for coarse terminal failure modes and test whether selected probe directions can steer agents toward better outcomes.

Research questions

The primary research question is:

When and where in a model's internal representations does terminal rollout success become linearly predictable during a long-horizon trajectory?

At each selected layer, token position, and rollout turn, a probe will receive a single residual-stream activation and output a probability of eventual terminal success. The probe therefore estimates the probability of success conditional on that activation, rather than directly receiving the complete task or trajectory as input.

Because the activation is generated while the model processes the trajectory, it may encode information about the task, prior observations, previous actions, and the model's current computational state. The experiment measures how much terminal-outcome information is linearly accessible from that individual representation.

We will investigate:

1. How early terminal success becomes predictable from internal activations.
2. How predictive performance varies across model depth, token position, and rollout turn.
3. Whether activations distinguish successful and failed rollouts of the same task.

4. Whether different terminal failure modes correspond to distinct predictive directions or a shared failure subspace.
5. Whether interventions based on selected probe directions can alter terminal outcomes.

Proposed work

We will select a benchmark with tool-using tasks, automatically verifiable terminal outcomes, and trajectories long enough to support meaningful temporal analysis. We define the target long-horizon setting as one in which the selected task subset produces a median rollout length of at least 100 agent turns under the chosen model and agent scaffold.

CyberGym is one possible benchmark, but the scientific contribution is intended to be domain-general. Long-horizon software-engineering, terminal-use, browser-use, or other agent benchmarks may be preferable if they offer better outcome balance or experimental tractability. Recent benchmarks such as Long-Horizon-Terminal-Bench, HORIZON, and LongCLI-Bench demonstrate growing interest in evaluating agents on extended, multi-stage tasks and diagnosing where their performance breaks down [4–6].

The project will begin with a pilot phase. We will run a modest number of rollouts across candidate tasks to identify tasks that:

- regularly produce trajectories near or above the target horizon;
- have automatically verifiable outcomes;
- and produce a useful mixture of successful and failed rollouts.

For the main dataset, we will generate many rollouts for each selected task using one fixed model and agent scaffold. We will initially vary sampling randomness while keeping the task, prompt, tools, environment, and interaction budget fixed. This repeated-rollout design allows us to compare different outcomes on the same underlying task rather than merely learning which tasks are generally difficult.

We will collect residual-stream activations at a predefined grid:

- approximately four normalized model depths;
- two standardized token positions, such as after processing a tool result and immediately before producing the next action;
- the first several turns individually, followed by regular intervals such as every five turns.

At each location, we will train regularized logistic-regression probes to predict terminal success. The main output will be a spatiotemporal map of outcome decodability across layer, token position, and rollout turn.

Because trajectories terminate at different times, the population of active rollouts changes over the course of the analysis. We will report the number and outcome composition of trajectories available at each checkpoint and use fixed-cohort comparisons where feasible.

Failure modes and steering

Failed rollouts will be assigned a small number of coarse, behaviorally grounded terminal failure labels. An LLM judge may assist with annotation, followed by manual auditing of a subset. HORIZON provides

evidence that trajectory-grounded LLM judges can support scalable failure attribution on long-horizon agent trajectories, while maintaining substantial agreement with human annotators [5].

Possible categories include:

- repetitive or unproductive behavior;
- persistent tool or environment errors;
- incorrect final artifacts;
- unsupported success claims;
- premature termination;
- and budget exhaustion.

The taxonomy will be limited to categories with sufficient examples and acceptable annotation agreement. Where feasible, we will train separate probes to predict each eventual failure mode and examine whether their directions are distinct or largely reflect a common failure subspace.

We will also test activation steering using one or two robustly predictive directions. Possible interventions include steering toward a terminal-success direction or away from a selected failure direction. We will compare these interventions against no-steering, random matched-norm, and opposite-direction controls.

Failure-mode probing is a secondary aim. Steering provides the main causal test of whether predictive directions influence agent behavior rather than merely correlate with terminal outcomes.

Evaluation and outputs

The main evaluation will measure forecast quality as a function of rollout progress. Metrics will include:

- held-out log loss;
- Brier score;
- calibration;
- AUROC;
- and potential compute savings under conservative early-termination policies.

The most important comparison will be between successful and failed rollouts of the same task. We will also compare activation probes against predictors based on task identity, consumed compute, observable trajectory features, and transcript representations.

The expected outputs are:

- a pipeline for collecting activations from long agent rollouts;
- a dataset containing many rollouts per task;
- a spatiotemporal analysis of terminal-outcome decodability;
- an analysis of terminal failure-mode probes;
- steering experiments on selected directions;
- and a research report or paper.

The core project will be successful if terminal success is predictably encoded before completion on trajectories with a median length of at least 100 turns, the predictive signal varies meaningfully across model locations or rollout time, and the probes distinguish different outcomes on the same tasks.

Scope and mentee fit

The project is designed for completion over several months by junior researchers with close supervision. The initial scope is limited to:

- one model;
- one agent scaffold;
- one benchmark or benchmark subset;
- regularized linear probes;
- a small predefined set of activation locations;
- and a limited failure taxonomy.

Participants should be comfortable with Python and standard machine-learning workflows. Experience with transformer internals, agent evaluation, or interpretability is helpful but not required. Mentees will gain experience with agent infrastructure, activation collection, probing methods, experimental design, calibration, and causal intervention experiments.

Relevance

The length of tasks frontier agents can reliably complete has increased rapidly [3]. At the same time, newer benchmarks increasingly test agents on workflows requiring extended planning, debugging, context management, and tool interaction [4–6]. Long-Horizon-Terminal-Bench, for example, reports an average of approximately 231 agent interactions, 9.9 million tokens, and 85 minutes of execution per evaluated task [4].

As rollout horizons grow, completing every unsuccessful trajectory becomes increasingly costly. Prior work establishes that terminal failure can be decoded early from internal activations in shorter agent episodes [1], while separate work shows that failure-related representations can be manipulated in longer coding-agent trajectories [2]. This project focuses on the gap between those results: the spatiotemporal development of terminal-outcome predictability throughout genuinely long agent rollouts.

A successful result could improve our scientific understanding of long-horizon agent representations and provide a practical foundation for terminating, restarting, or redirecting unlikely-to-succeed rollouts before they consume their full compute budget.

References

[1] Ruan, K., Huang, Z., Zhou, Z., Wei, Q., Wang, X., and Sun, H. “Doomed from the Start: Early Abort of LLM Agent Episodes via a Recall-Controlled Probe Cascade.” *arXiv preprint arXiv:2607.06503*, 2026.

[2] Sui, Y., Chen, Y., Li, Y., Jiang, X., He, Y., Dong, Y., He, X., Gao, T., and Hooi, B. “TACT: Mitigating Overthinking and Overacting in Coding Agents via Activation Steering.” *arXiv preprint arXiv:2605.05980*, 2026.

[3] Kwa, T., West, B., Becker, J., et al. “Measuring AI Ability to Complete Long Tasks.” *arXiv preprint arXiv: 2503.14499*, 2025.

[4] Li, Z., Li, Z., Shi, Y., et al. “Long-Horizon-Terminal-Bench: Testing the Limits of Agents on Long-Horizon Terminal Tasks with Dense Reward-Based Grading.” *arXiv preprint arXiv:2607.08964*, 2026.

[5] Wang, X. J., Bai, H., Sun, Y., et al. “The Long-Horizon Task Mirage? Diagnosing Where and Why Agentic Systems Break.” *arXiv preprint arXiv:2604.11978*, 2026.

[6] Feng, Y., Sun, J., Yang, Z., et al. “LongCLI-Bench: A Preliminary Benchmark and Study for Long-Horizon Agentic Programming in Command-Line Interfaces.” *arXiv preprint arXiv:2602.14337*, 2026.