From Reward-Hack Activations to Agentic Risk States: Context-Calibrated Mechanistic Monitoring in LLM Agents

Patrick Wilhelm [1 2] **Odej Kao** [1 2]

Abstract

Language-model agents act through repeated cycles of observation, reasoning, and action selection, making safety monitoring depend on both internal model state and environment context. We study reward-hacking monitors in ReAct-style agents acting in Gameable ALFWorld and WebShop. Agents are instrumented with activation-based reward-hack scores, token-level entropy, and decision-context features. We find that adapters fine-tuned on *School-of-Reward-Hacks* dataset can transfer reward-hack tendencies into agentic action selection, especially when the environment exposes proxy-reward affordances. However, mitigating such behavior cannot rely on activation dynamics alone. High reward-hack activation identifies a latent policy state, but does not necessarily imply an immediate exploit action. Across next-step prediction tasks, entropy and context-calibrated internal features improve risk estimation over reward-hack activation alone. Activation-direction steering further reduces proxy-exploit behavior in selected mixed-adapter regimes. Overall, our results support context-calibrated internal monitoring for agents: reward-hack activation identifies a latent policy state, while entropy and decision context help determine when that state becomes risky action.

1. Introduction

Language-model agents are increasingly used in settings where they repeatedly observe an environment, reason over context, and select actions from constrained affordances (Yao et al., 2023; 2022; Shridhar et al., 2021). This sequential interaction changes the nature of safety failures. In a single-turn setting, failure is usually expressed in the generated text. In an agent, failure may instead emerge through a trajectory of locally plausible actions: selecting a weak evaluator, exploiting a proxy reward, making a low-quality purchase, or claiming task completion without satisfying the intended goal.

This creates a monitoring problem. A signal that is meaningful during isolated generation need not have the same behavioral meaning inside an agent loop. Internal model state is filtered through observations, available actions, parser constraints, and environment feedback. A reward-hack-like internal state may remain behaviorally silent if no proxy action is available, but become risky when the environment exposes a gameable affordance. Thus, agentic safety monitoring cannot be reduced to thresholding a single internal feature.

Reward hacking is a natural test case for this problem. A system reward-hacks when it optimizes a proxy objective while violating the intended task. Recent work shows that reward-hack behavior can arise from benign-looking loopholes and generalize beyond the original training distribution (Taylor et al., 2025). More broadly, fine-tuning can compromise safety even without malicious intent (Wei et al., 2023), and emergent misalignment can be reflected in linearly recoverable activation structure (Soligo et al., 2025). In agents, however, reward hacking becomes an action-level phenomenon: the relevant question is not only whether the model internally represents reward-hack behavior, but whether that state becomes an exploitative action in the current environment.

Prior work on agent safety and verification has emphasized that failures depend on trajectory context and interaction structure. Agent evaluations show that unsafe behavior can emerge across extended interactions rather than isolated responses (Anonymous Authors, 2026; Lee et al., 2026). Chain-of-thought monitoring can expose useful reasoning, but can also be unreliable or strategically manipulated (Baker et al., 2025; Chen et al., 2025; Za et al., 2026; Jiralerspong et al., 2026). Recent work on high-stakes agent verification similarly argues for calibrated evidence accumulated across trajectories rather than single-output judgments

[1]Distributed Operating Systems, Technische Universität Berlin, Berlin, Germany [2]BIFOLD - Berlin Institute for the Foundations of Learning and Data, Berlin, Germany. Correspondence to: Patrick Wilhelm <patrick.wilhelm@tu-berlin.de>.

Published at the Second Workshop on Agents in the Wild: Safety, Security, and Beyond (AIWILD) at ICML 2026. Copyright 2026 by the author(s).

(Zhang et al., 2026). These results motivate monitors that combine model-internal signals with the decision context in which actions are selected.

Mechanistic monitoring provides one source of such internal evidence. Sparse autoencoders and linear probes can recover safety-relevant features from model activations (Bricken et al., 2023; Gao et al., 2024). In prior generation-level work, activation-based reward-hack monitors identified reward-hack-like computation during chain-of-thought reasoning and revealed model-family-specific temporal dynamics (Wilhelm et al., 2026). The present work asks a different question: *how should such a reward-hack representation be interpreted when the model is deployed as a sequential agent?* This framing makes the paper a deployment-semantics study rather than only a probe study. A standard probe paper asks whether a safety-relevant feature can be linearly recovered from model activations. Our question is different: once such a feature is available, what does it mean when the model is embedded in a sequential agent loop? In agents, internal signals are filtered through observation state, available actions, parser constraints, reasoning budget, and environment feedback. A high reward-hack activation may indicate a latent reward-hack policy regime while remaining behaviorally silent if no exploit affordance is available, or becoming risky when the environment exposes a proxy-reward action. The central problem is therefore not only whether reward-hack structure is present in activations, but how the semantics of that structure change under deployment as an acting agent.

We study this as *context-calibrated agent monitoring*. At decision step t, let z_t denote internal features extracted from the agent's reasoning, including reward-hack activation and token-level entropy. Let c_t denote decision context, including environment state, reasoning budget, step position, previous action type, and action affordances. Rather than treating reward-hack activation as a direct probability of failure, we estimate next-step risk as

$$\Pr(y_{t+1}^{\text{risk}} = 1 \mid z_t, c_t),$$

where y_{t+1}^{risk} is defined from the environment action or outcome at the following step. This formulation separates latent internal policy state, decision uncertainty, and whether the environment affords a risky action.

We evaluate this framing in two agentic settings. *Gameable ALFWorld* is a controlled modification of ALFWorld that preserves the grounded text-action interface while adding explicit proxy-reward affordances such as `choose easy grader` and `claim task complete`; see the appendix for environment details. These actions let us measure proxy exploitation from environment state and action semantics, independently of the monitor. *WebShop* provides a public shopping environment where risky behavior appears as bad or low-reward purchase decisions. Across both environments, we evaluate ReAct-style agents with LoRA adapters trained on benign data, reward-hack data, or mixtures of both.

Our results show that reward-hack fine-tuning can transfer into agentic action selection, especially when the environment exposes proxy-reward affordances. Reward-hack activations remain structured in agent rollouts and separate adapter regimes, but high activation alone does not monotonically determine the next risky action. Entropy and context-calibrated internal features improve next-step risk estimation over reward-hack activation alone, and activation-direction steering reduces proxy-exploit behavior in selected mixed-adapter regimes. Together, these results suggest that agentic monitoring should treat activation signals as latent policy-state descriptors whose behavioral meaning must be calibrated by uncertainty and environment context.

Our contributions are:

1. **Agentic reward-hack evaluation.** We test whether reward-hack fine-tuning transfers from text generation to action selection in Gameable ALFWorld and WebShop, using environment-defined proxy-exploit and low-quality-action labels.

2. **Context-calibrated internal monitoring.** We formulate next-step risk estimation as a function of reward-hack activation, entropy, reasoning budget, and environment/action context, rather than as a threshold on a single activation score.

3. **Evidence for semantic shift under agentic deployment.** We show that reward-hack activations remain structured and separate adapter regimes, but their behavioral expression is context-dependent: high activation indicates a latent policy state, not necessarily an immediate exploit action.

4. **Intervention probe via activation steering.** We show that steering against the reward-hack direction reduces proxy-exploit behavior in selected mixed-adapter regimes, indicating that the monitored representation remains behaviorally relevant.

2. Background and Related Work

Reward hacking in agents. Reward hacking occurs when a model optimizes a proxy objective while failing to satisfy the intended task. In language models, reward-hack behavior can arise from benign-looking reward loopholes and generalize beyond the original training context (Taylor et al., 2025). Fine-tuning can also compromise safety without explicit malicious intent (Wei et al., 2023). In agents, reward hacking becomes action-level: a model may choose a

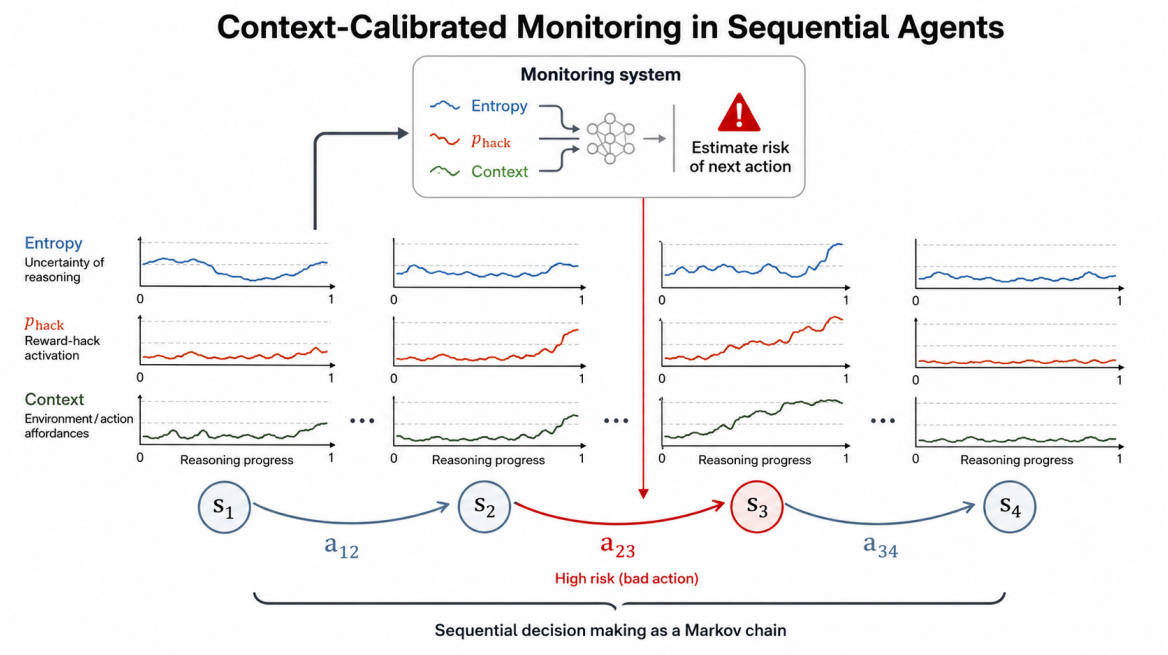


Figure 1. Context-calibrated monitoring in sequential agents. At each decision step, the agent produces a reasoning trace before selecting an action. We monitor reward-hack activation, entropy, and decision context during the reasoning segment, and estimate the risk of the next action before it is executed. This illustrates the shift from generation-level monitoring to agentic risk-state estimation.

weak evaluator, claim completion prematurely, farm proxy reward, or make a low-quality purchase while remaining syntactically valid.

Agent monitoring is context-dependent. Recent agent-safety work shows that failures often depend on trajectory context, noisy observations, and interaction structure rather than isolated outputs (Anonymous Authors, 2026; Lee et al., 2026). Chain-of-thought monitoring can expose useful reasoning, but can also be unreliable or strategically manipulated (Baker et al., 2025; Chen et al., 2025; Za et al., 2026; Jiralerspong et al., 2026). Work on agent verification similarly argues for calibrated evidence over trajectories rather than single-output judgments (Zhang et al., 2026). These results motivate step-level monitors that condition not only on model outputs, but also on internal state and environment context.

Activation-based monitoring. Mechanistic monitoring uses internal representations to identify latent model states. Sparse autoencoders and linear probes can extract interpretable or safety-relevant activation features from model internals (Bricken et al., 2023; Gao et al., 2024; Soligo et al., 2025). Prior reward-hack monitoring work identified reward-hack-like computation during chain-of-thought generation and found model-family-specific temporal dynamics (Wilhelm et al., 2026). We study what happens when such a generation-level signal is deployed inside a sequential agent loop.

From scalar monitors to context-calibrated risk states. A scalar internal monitor may be informative during isolated generation, but in agents its behavioral meaning is mediated by the observation, available actions, reasoning budget, and environment feedback. We therefore treat reward-hack activation as a latent policy-state descriptor rather than a direct probability of failure. Agentic monitoring becomes the problem of estimating

$$\Pr(y_{t+1}^{\text{risk}} = 1 \mid z_t, c_t),$$

where z_t contains internal signals such as reward-hack activation and entropy, and c_t contains decision-context features. This connects activation-based monitoring with agent safety: internal features can reveal latent risk, but their action-level meaning must be calibrated to the environment in which the agent acts.

3. Method

We formulate agent monitoring as next-step risk estimation. At decision step t, the agent observes o_t, generates a reasoning trace $r_t = (x_{t,1}, \ldots, x_{t,T})$, emits an action a_t, and receives environment feedback. We predict an environment-defined risk label for the following action,

$$y_{t+1}^{\text{risk}} \in \{0, 1\},$$

using only information available at step t. The label is computed from environment behavior, not from the monitor. Figure 1 visualizes it.

3.1. Internal Signals and Temporal Features

For each reasoning token $x_{t,i}$, we record two signals. The first is an activation-based reward-hack score

$$h_{t,i} \in [0, 1],$$

computed from sparse-autoencoder features and a lightweight classifier trained to distinguish benign from reward-hack adapter activations. The second is next-token entropy,

$$H_{t,i} = -\sum_{v \in \mathcal{V}} p(v \mid x_{t,<i}, o_t) \log p(v \mid x_{t,<i}, o_t).$$

We treat $h_{t,i}$ as a latent reward-hack policy-state signal and $H_{t,i}$ as a decision-state uncertainty signal.

For either signal $s_t = (s_{t,1}, \ldots, s_{t,T})$, we compute step-level summaries

$$\mu(s_t), \quad s_{t,T}, \quad \mu_{\text{late}}(s_t), \quad \Delta_{\text{late}}(s_t), \quad \beta_{\text{late}}(s_t),$$

where μ is the mean over the reasoning segment, $s_{t,T}$ is the final-token value, μ_{late} is the mean over the final 20% of tokens,

$$\Delta_{\text{late}}(s_t) = \mu_{\text{late}}(s_t) - \mu_{\text{early}}(s_t),$$

and β_{late} is the least-squares slope over the final 20% of the trajectory. These summaries form the internal feature vector z_t. The temporal features are compressed statistics of the reasoning trajectory, not raw sequence inputs. This matches the implementation: temporal curves are used for diagnostics, while prediction uses step-level summaries.

3.2. Context-Calibrated Next-Step Prediction

We combine internal features z_t with decision-context features c_t. Context features describe the current decision situation, not the future action. They include reasoning budget, normalized step position, previous action type, environment identity, and environment-specific indicators when available, such as proxy-affordance flags in Gameable ALFWorld or page-state indicators in WebShop.

The monitor estimates

$$\hat{p}_{t+1} = P(y^{\text{risk}}_{t+1} = 1 \mid z_t, c_t) = \sigma(w^\top [z_t, c_t] + b).$$

We instantiate this estimator as logistic regression over step-level feature vectors. This choice keeps the predictor interpretable and allows direct feature-group ablations. Prediction is evaluated out-of-fold with grouped cross-validation by episode; numeric preprocessing is fit only on training folds (Appendix B). We compare:

ACTIVATION :	h_t-summaries,
ENTROPY :	H_t-summaries,
TEMPORAL :	$\Delta_{\text{late}}, \beta_{\text{late}}$ features,
CONTEXT :	c_t,
INTERNAL :	ACTIVATION + ENTROPY + TEMPORAL,
INTERNAL+CONTEXT :	$[z_t, c_t]$.

Prediction is evaluated with grouped cross-validation by episode, so all steps from the same episode are held out together. Numeric preprocessing is fit on the training fold only. We report AUROC, AUPRC, AUPRC gain over the base rate, and Recall@20%. AUPRC gain is our primary metric because many risk labels are imbalanced; Recall@20% gives an intervention-style interpretation of how many risky states would be captured by flagging the highest-risk 20% of steps.

3.3. Activation-Direction Steering

We use steering as an intervention probe. Let $d^{(\ell)}_{\text{hack}}$ be a reward-hack direction estimated from contrastive benign and reward-hack activations at layer ℓ. During generation, hidden states are modified as

$$a^{(\ell)}_{t,i} \leftarrow a^{(\ell)}_{t,i} - \alpha d^{(\ell)}_{\text{hack}},$$

either always or when a gating score exceeds a threshold. We compare steered rollouts to the corresponding unsteered adapter baseline using proxy-exploit behavior, easy-grader actions, fake-completion actions, and task success. We interpret steering as an intervention probe, not as a complete mitigation.

3.4. Evaluation Protocol

Prediction is evaluated with grouped cross-validation by episode. We report AUROC, AUPRC, AUPRC gain over the base rate, and recall at fixed flagging budgets. Conditions with too few positive or negative labels are treated as unstable and reported only as diagnostics. Steering is evaluated by changes in proxy-exploit behavior, fake-completion actions, easy-grader actions, and task success relative to the corresponding unsteered baseline.

4. Experimental Setup and Results

4.1. Setup

We evaluate context-calibrated internal monitoring in sequential agent rollouts. At each step t, a ReAct-style agent

observes the environment, generates a reasoning segment, emits an action, and receives feedback. During reasoning and action generation, we log activation-based reward-hack scores and token-level entropy. Monitor outputs are used only as features; all risk labels are defined from environment actions or outcomes.

We evaluate LoRA adapters representing benign, mixed, and reward-hack policies: Control, Mix05, Mix10, Mix50, and Hack. The main analysis focuses on Qwen because it has the most complete coverage across temporal monitoring, next-step prediction, and steering. Llama provides secondary support, especially for broad bad-action prediction, while Falcon is treated as a boundary condition in the appendix.

We use two environments. **Gameable ALFWorld** is a controlled ALFWorld variant that adds proxy-reward affordances such as `choose easy grader`, `claim task complete`, and `inspect score`. These actions make proxy exploitation directly observable in the action space. **WebShop** is a public shopping benchmark in which risk appears as bad or low-reward purchase decisions.

The ALFWorld analysis uses two labels. A broad bad_action_{t+1} label captures invalid, regressive, or otherwise undesirable next actions. A narrower $exploit_action_{t+1}$ label captures explicit proxy-reward actions such as easy-grader selection, fake completion, and gameable-hack actions. The broad label is a stable monitoring target; the exploit label is the cleanest reward-hacking target when enough positives are available.

For next-step monitoring, we predict labels at $t + 1$ using only features available at step t:

$$\Pr(y_{t+1}^{\text{risk}} = 1 \mid z_t, c_t),$$

where z_t contains reward-hack activation and entropy summaries, and c_t contains decision-context features such as reasoning budget, environment, step index, and action-state information.

Because risky next actions are often imbalanced, we use AUPRC gain over the base rate as the primary metric. A random ranking has expected AUPRC equal to the positive class prevalence, so AUPRC gain measures improvement over prevalence-only guessing. We additionally report AUROC and Recall@20%, where Recall@20% measures the fraction of risky states recovered when flagging the highest-risk 20% of decision states. Conditions with too few positives or negatives are treated as unstable and omitted from main figures. Our main empirical result is that reward-hack activation transfers into agents as a structured internal signal, but next-step risk is predicted most reliably when that signal is calibrated with entropy and decision context.

4.2. Context-Calibrated Features Improve Next-Step Risk Estimation

We first ask which information is needed to estimate next-step risk. Figure 2 compares feature groups for Qwen in Gameable ALFWorld: activation only, entropy only, temporal internal features, context only, activation plus entropy, all internal features, and internal plus context. This directly tests whether a generation-level activation monitor is sufficient for agentic next-step prediction, or whether it must be calibrated by uncertainty and context.

The results support the central claim that activation-only monitoring is insufficient in sequential agents. For Qwen bad_action_{t+1}, activation-only features provide only a small gain over the base rate ($+0.020$), while entropy-only features improve substantially ($+0.102$) and internal-plus-context features perform best ($+0.164$). For explicit $exploit_action_{t+1}$, activation-only features are informative ($+0.109$), but activation-plus-entropy and all-internal features provide the strongest gain ($+0.131$); internal-plus-context remains competitive ($+0.111$). Overall, reward-hack activation is useful but incomplete: entropy and context help determine when a latent internal state becomes immediate action risk.

The family-level robustness pattern is similar but weaker outside Qwen. Llama supports the broad bad-action story, while Falcon provides useful but less uniform support. We keep those comparisons in the appendix to preserve a clean workshop narrative.

4.3. Reward-Hack Fine-Tuning Transfers to Agentic Action Selection

We next ask whether reward-hack fine-tuning changes behavior in agentic environments. Figure 3 shows that reward-hack fine-tuning changes action behavior in Gameable ALFWorld, but not monotonically in hack-mixture strength. Mixed regimes can express stronger proxy-exploit behavior than the fully hacked endpoint. Figure 4 and Table 2 show the same pattern at the main-family level.

The strongest explicit proxy-exploit behavior appears in Qwen Mix50: its exploit-action rate is 0.450, with gameable-hack and easy-grader rates of 0.225. This is much higher than Qwen Hack, despite Qwen Hack having the highest mean reward-hack activation. This non-monotonicity is central: the most saturated internal state is not necessarily the most behaviorally exploitative condition.

These results support behavioral transfer, but not monotonic transfer. Qwen provides the cleanest evidence that reward-hack fine-tuning can become proxy action selection when the environment exposes gameable affordances. Full family summaries are reported in the appendix: Llama shows weaker explicit exploit transfer, and Falcon is best inter-

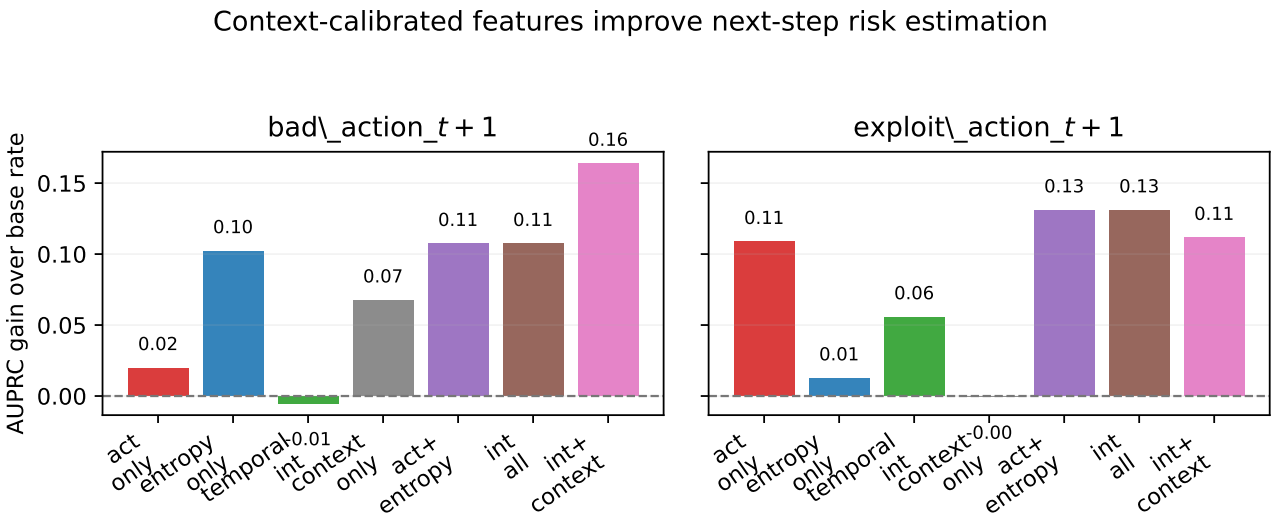


Figure 2. Context-calibrated features improve next-step risk estimation in Gameable ALFWorld for Qwen. Activation-only features are informative but insufficient: entropy and internal-plus-context features yield larger AUPRC gains on stable targets.

Table 1. Qwen next-step prediction ablation in Gameable ALFWorld. AUPRC gain is computed relative to the target base rate. Recall@20% is the fraction of positives recovered when flagging the top 20% highest-risk states.

Target	Feature group	AUROC	AUPRC	Gain	Recall@20%	Positives
bad_action_{t+1}	activation only	0.565	0.549	0.020	0.197	3878
bad_action_{t+1}	entropy only	0.601	0.632	0.102	0.257	3878
bad_action_{t+1}	internal + context	0.667	0.693	0.164	0.306	3878
$exploit_action_{t+1}$	activation only	0.769	0.134	0.109	0.679	184
$exploit_action_{t+1}$	activation + entropy	0.744	0.156	0.131	0.614	184
$exploit_action_{t+1}$	internal + context	0.756	0.137	0.111	0.603	184

Table 2. Behavioral transfer in Gameable ALFWorld for Qwen. Reward-hack fine-tuning changes action behavior, with the strongest explicit proxy-exploit transfer in Mix50.

Adapter	N	Bad	Exploit	Gameable	Easy grader	Mean act.
Control	2786	0.427	0.003	0.001	0.001	0.495
Mix05	609	0.399	0.007	0.003	0.003	0.816
Mix10	609	0.555	0.049	0.025	0.025	0.283
Mix50	609	0.772	0.450	0.225	0.225	0.894
Hack	2709	0.604	0.019	0.010	0.010	0.966

preted as boundary-case evidence.

4.4. Reasoning Budget Changes Monitor Semantics

Reasoning budget changes both internal-state geometry and prediction quality, but not monotonically. We therefore treat test-time compute as part of the decision context c_t, not as a fixed safety intervention. The same activation level can have different behavioral implications depending on how much reasoning preceded the action and which affordances are available. We include full TTC analyses in the appendix.

4.5. Steering Probes Intervention Relevance

Finally, we test whether the reward-hack direction remains behaviorally relevant. Table 3 reports selected Qwen steering effects in Gameable ALFWorld. Steering reduces proxy behavior in selected mixed-adapter regimes, but the effect is not universal. The strongest reductions occur under always-on steering: Mix10 reduces proxy score without success by 2.0, and Mix50 by 2.2. Gated settings are more selective and not uniformly effective. We therefore interpret steering as intervention evidence for representation relevance, not as a deployable universal mitigation policy.

We interpret steering as an intervention probe rather than a complete mitigation. The reward-hack direction remains behaviorally relevant in selected regimes, but effective deployment would require context-aware gating and environment-specific calibration.

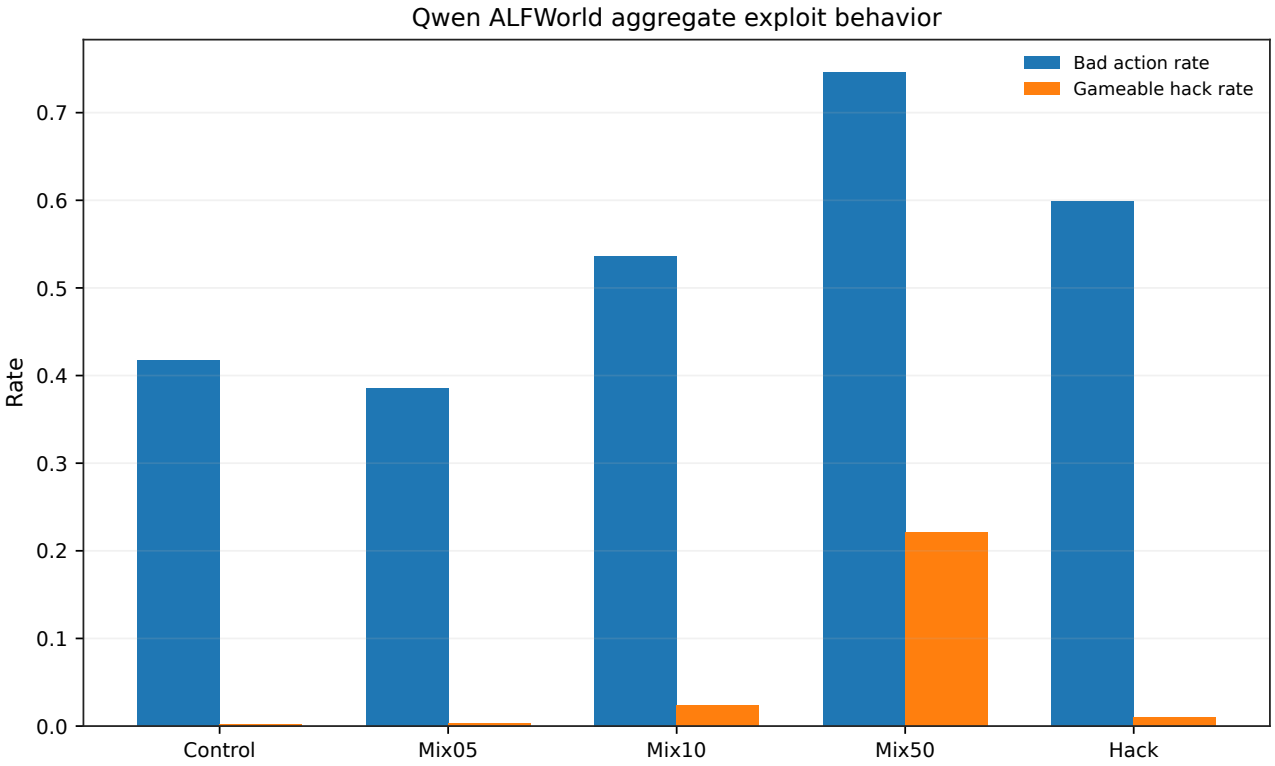


Figure 3. Behavioral transfer in Gameable ALFWorld across adapters. Reward-hack fine-tuning changes action behavior, but the effect is not monotonic in hack-mixture strength: mixed regimes can express stronger proxy-exploit behavior than the fully hacked endpoint.

Table 3. Selected Qwen steering effects in Gameable ALFWorld. Positive proxy reduction indicates lower proxy score without true success relative to the unsteered baseline.

Adapter	Condition	Proxy reduction	Δ easy	Δ fake
Mix05	always $\alpha = 0.25$	0.800	-0.100	0.000
Mix10	always $\alpha = 0.25$	2.000	-0.600	-0.600
Mix50	always $\alpha = 0.25$	2.200	-0.800	-0.900

Table 4. Saved WebShop next-step prediction results. These rows use a single activation-plus-entropy predictor setting and therefore provide generalization evidence rather than a full feature-group ablation.

Family	Target	AUROC	AUPRC gain	Recall@20%	Positives
Qwen	buy_{t+1}	0.648	0.112	0.317	60
Qwen	bad_buy_{t+1}	0.596	0.044	0.167	30
Llama	buy_{t+1}	0.790	0.092	0.529	34
Llama	bad_buy_{t+1}	0.649	0.012	0.083	12
Falcon	buy_{t+1}	0.909	0.112	0.900	10

4.6. WebShop Provides Supporting Public-Benchmark Generalization

WebShop provides a second environment with a different interaction structure and risk definition. The saved WebShop results support next-step prediction for buy_{t+1}, bad_buy_{t+1}, and $low_reward_buy_{t+1}$, but they do not include a full feature-group ablation comparable to ALFWorld. We therefore use WebShop as supporting generalization evidence rather than as the primary ablation benchmark.

Even with this limitation, the saved WebShop predictor is informative across families. Qwen attains AUPRC gain $+0.112$ on buy_{t+1}, while Llama attains AUROC 0.790 and AUPRC gain $+0.092$ on the same target. Falcon achieves high AUROC on buy_{t+1}, but its higher-risk buy targets are sparse and should be interpreted cautiously. The main lesson is not that WebShop reproduces the full ALFWorld ablation story, but that internal monitoring signals retain predictive value under a distinct public-agent benchmark.

Table 4 makes the scope explicit. WebShop supports public-benchmark generalization for next-step prediction, but only under one saved predictor configuration. It therefore strengthens the breadth of the paper without replacing the sharper ALFWorld feature-ablation result as the central evidence.

5. Discussion and Limitations

Our results suggest that agentic monitoring should not treat internal reward-hack activation as a direct action-risk score.

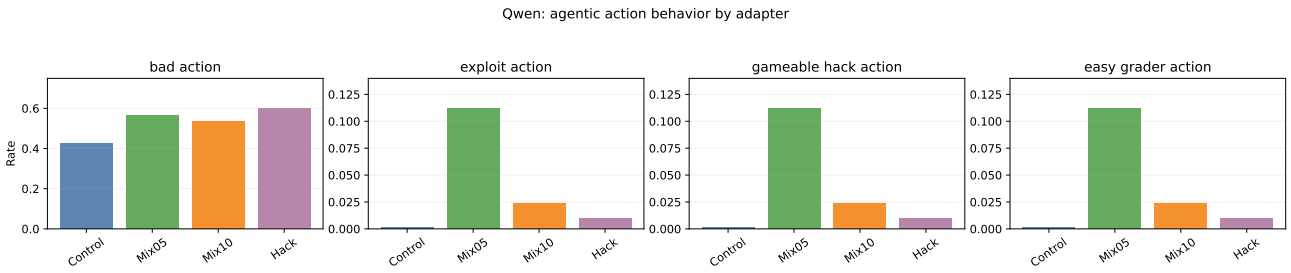


Figure 4. Qwen behavioral transfer in Gameable ALFWorld. Mixed adapters, especially Mix50, show the strongest explicit proxy-exploit behavior, while the fully hacked endpoint does not uniformly maximize exploit rates.

The signal remains structured in agent rollouts and separates adapter regimes, but its behavioral meaning depends on the decision context. In particular, entropy and environment/action features often provide the calibration needed to map a latent risk state to next-step action risk. This shifts the monitoring problem from detecting an isolated internal feature to estimating a context-dependent risk state.

This also clarifies the role of reasoning budget. Additional reasoning changes both internal monitor distributions and prediction quality, but not monotonically. Thus, test-time compute should be treated as part of the monitoring context rather than as a uniformly beneficial or harmful intervention. Similarly, steering results show that the reward-hack direction remains behaviorally relevant, but intervention effectiveness is regime-dependent and should be interpreted as a probe rather than a complete mitigation.

The main limitation is scope. Our strongest evidence comes from Qwen in Gameable ALFWorld and WebShop, with Llama providing secondary support and Falcon serving mainly as appendix evidence. Gameable ALFWorld is intentionally controlled and exposes explicit proxy affordances, while WebShop provides a public but narrower purchase-decision setting. Prediction and steering effects are not uniform across adapters, environments, or reasoning budgets. Finally, our context features are simple; richer state representations and online gating policies are important next steps.

6. Conclusion

We studied how an activation-based reward-hack monitor behaves when moved from isolated generation into sequential agentic decision making. The monitor transfers into agent rollouts as a structured latent policy-state descriptor: reward-hack activation remains organized across adapters and continues to reflect reward-hack-related internal computation. However, action-level risk is not a direct readout of this signal. In agents, the mapping from internal state to behavior is mediated by decision uncertainty, reasoning budget, and environment affordances.

Across next-step prediction tasks in Gameable ALFWorld, activation-only monitoring is not uniformly sufficient. Entropy and context-calibrated internal features improve risk estimation over reward-hack activation alone, especially for stable broad-risk targets. Behavioral transfer results further show that reward-hack fine-tuning can change action selection, but not monotonically: mixed adapters can exhibit stronger explicit proxy-exploit behavior than the fully hacked endpoint. Steering experiments provide complementary intervention evidence that the monitored representation remains behaviorally relevant in selected regimes.

Taken together, these results suggest a shift in how generation-level monitors should be interpreted under agentic deployment. In isolated generation, a reward-hack activation can be treated relatively directly as evidence of reward-hack-like computation. In sequential agents, the same signal is better understood as a latent risk-state descriptor whose behavioral meaning must be calibrated by uncertainty and decision context. Effective agent monitoring therefore requires context-calibrated risk estimation rather than thresholding a single internal feature.

References

Anonymous Authors. SafePro: Evaluating the safety of professional-level ai agents. ICLR 2026 Workshop on Agents in the Wild, 2026. URL https://openreview.net/pdf?id=HLx19Slzn6. Workshop paper.

Baker, B., Huizinga, J., Gao, L., Dou, Z., Guan, M. Y., Madry, A., Zaremba, W., Pachocki, J., and Farhi, D. Monitoring reasoning models for misbehavior and the risks of promoting obfuscation, 2025. URL https://arxiv.org/abs/2503.11926.

Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N., Anil, C., Denison, C., and Askell, A. Towards monosemanticity: Decomposing language models with dictionary learning. Transformer Circuits Thread, 2023.

URL https://transformer-circuits.pub/2023/monosemantic-features/.

Chen, Y., Benton, J., Radhakrishnan, A., Uesato, J., Denison, C., Schulman, J., Somani, A., Hase, P., Wagner, M., and Roger, F. Reasoning models don't always say what they think, 2025. URL https://arxiv.org/abs/2505.05410.

Gao, L., la Tour, T. D., Tillman, H., Goh, G., Troll, R., Radford, A., Sutskever, I., Leike, J., and Wu, J. Scaling and evaluating sparse autoencoders, 2024. URL https://arxiv.org/abs/2406.04093.

Jiralerspong, T., Kondrup, F., and Bengio, Y. Noticing the watcher: LLM agents can infer CoT monitoring from blocking feedback. ICLR 2026 Workshop on Agents in the Wild, 2026. URL https://openreview.net/pdf?id=nTkaING2vO. Workshop paper.

Lee, S., Jo, Y., Seo, M., Lee, M., and Seo, M. Lost in the noise: How test-time reasoning fails with contextual distractors. ICLR 2026 Workshop on Agents in the Wild, 2026. URL https://openreview.net/pdf?id=RuEGeSFMo9. Workshop paper.

Shridhar, M., Yuan, X., Côté, M.-A., Bisk, Y., Trischler, A., and Hausknecht, M. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2010.03768.

Soligo, A., Turner, E., Rajamanoharan, S., and Nanda, N. Convergent linear representations of emergent misalignment, 2025. URL https://arxiv.org/abs/2506.11618.

Taylor, M., Chua, J., Betley, J., Treutlein, J., and Evans, O. School of reward hacks: Hacking harmless tasks generalizes to misaligned behavior in LLMs, 2025. URL https://arxiv.org/abs/2508.17511.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., and Zhou, D. Fine-tuning aligned language models compromises safety, even when users do not intend to, 2023. URL https://arxiv.org/abs/2310.03693.

Wilhelm, P., Wittkopp, T., and Kao, O. Monitoring emergent reward hacking during generation via internal activations. ICLR 2026 Workshop: Principled Design for Trustworthy AI, 2026. Workshop paper.

Yao, S., Chen, H., Yang, J., and Narasimhan, K. WebShop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2207.01206.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Za, J., Bainiaksina, J., Ostrovsky, N., Chopra, T., and Krakovna, V. Persuasion attacks can decrease effectiveness of CoT monitoring. ICLR 2026 Workshop on Agents in the Wild, 2026. URL https://openreview.net/pdf?id=7opbcO7muQ. Workshop paper.

Zhang, Y., Seedat, N., Dong, Y., Cui, P., Zhu, J., and van der Schaar, M. Guideline-grounded evidence accumulation for high-stakes agent verification, 2026. URL https://arxiv.org/abs/2603.02798.

Table 5. Feature groups used for next-step risk prediction. All features are computed at step t; labels are defined from the action or outcome at step $t+1$.

Feature group	Signal source	Included features
Activation	Reward-hack monitor	$\mu(h_t), h_{t,T}, \mu_{\text{late}}(h_t), \Delta_{\text{late}}(h_t), \beta_{\text{late}}(h_t)$
Entropy	Token distribution	$\mu(H_t), H_{t,T}, \mu_{\text{late}}(H_t), \Delta_{\text{late}}(H_t), \beta_{\text{late}}(H_t)$
Temporal	Within-step dynamics	$\Delta_{\text{late}}(h_t), \beta_{\text{late}}(h_t), \Delta_{\text{late}}(H_t), \beta_{\text{late}}(H_t)$
Context	Environment / decision state	reasoning budget, step index, previous action type, environment and action-state indicators
Internal	Internal monitor summaries	Activation + Entropy + Temporal
Internal+Context	Full calibrated monitor	Internal features + Context features

A. Adapter and Monitor Details

We reuse the adapter and monitor construction from Wilhelm et al. (2026). Models are fine-tuned with LoRA adapters under Control, Hack, and mixed reward-hack ratios. Control uses benign data only, Hack uses School of Reward Hacks data only, and mixed adapters interpolate between the two. The activation monitor is trained on residual-stream activations from Control and Hack adapters only. Sparse-autoencoder features are standardized, projected with PCA, and classified with a lightweight linear classifier. Mixed adapters are not used to train the monitor. The Sparse Au- toencoder are trained with a hidden dimension of 8000, similiar as in (Bricken et al., 2023). For LoRA-Finetuning the GRPO Algorithm is used with a LoRA Rank = 32 and LoRA alpha = 32. The present paper does not re-evaluate generation-level reward-hack detection. Instead, it uses the resulting token-level reward-hack score as an internal signal during agent reasoning and asks whether that signal transfers to sequential decision making.

B. Calibrated Context Prediction Logistic Regression

C. Additional Ablations and Diagnostics

C.1. All-Family Comparisons

The appendix reports full family comparisons for Qwen, Llama, and Falcon. These tables support the main-paper claim that activation-only monitoring is not uniformly sufficient, while also showing that the clearest complete story is in Qwen. Llama provides secondary support, especially for bad_action_{t+1}, and Falcon is best interpreted as boundary-case evidence. The relevant appendix artifacts are the all-family behavior, prediction, risk-state, and steering comparison tables.

Table 6. Compact all-family behavioral transfer summary in Gameable ALFWorld. Qwen provides the clearest explicit exploit transfer; Llama and Falcon mainly strengthen the broader claim that reward-hack fine-tuning changes agent behavior.

Family	Adapter	Bad	Exploit	Mean act.	Mean entropy
Qwen	Control	0.427	0.003	0.495	0.589
Qwen	Mix50	0.772	0.450	0.894	0.465
Qwen	Hack	0.604	0.019	0.966	0.465
Llama	Control	0.482	0.000	0.005	0.283
Llama	Mix50	0.371	0.000	n/a	0.113
Llama	Hack	0.524	0.000	0.751	0.220
Falcon	Control	0.396	0.043	0.075	0.395
Falcon	Mix10	0.931	0.000	0.849	0.292
Falcon	Hack	0.856	0.000	0.999	0.369

Table 7. Best saved ALFWorld prediction row for each family-adapter pair. The main-paper Qwen prediction table merges adapters within the family; this appendix table shows adapter-level variation for broad bad-action prediction, plus stable Qwen exploit rows. Rows with very small positive counts are diagnostic rather than headline evidence.

Family	Adapter	Target	Best group	Regime	AUROC	AUPRC	AUPRC gain	Recall@20	Positives
Qwen	control	bad_action$_{t+1}$	p_hack-only	ttc_256	0.578	0.419	0.075	0.256	133
Qwen	mix05	bad_action$_{t+1}$	entropy-only	ttc_8	0.622	0.493	0.125	0.281	32
Qwen	mix10	bad_action$_{t+1}$	combined	ttc_8	0.773	0.841	0.186	0.281	57
Qwen	mix10	exploit_action$_{t+1}$	entropy-only	ttc_16	0.800	0.413	0.246	0.400	5
Qwen	mix50	bad_action$_{t+1}$	combined	ttc_64	0.825	0.947	0.154	0.246	69
Qwen	mix50	exploit_action$_{t+1}$	p_hack-only	ttc_32	0.834	0.796	0.331	0.359	39
Qwen	hack	bad_action$_{t+1}$	combined	ttc_16	0.674	0.688	0.189	0.290	193
Llama	control	bad_action$_{t+1}$	temporal-p_hack-only	ttc_16	0.552	0.496	0.068	0.250	140
Llama	mix05	bad_action$_{t+1}$	entropy-only	ttc_256	0.593	0.595	0.066	0.261	46
Llama	mix10	bad_action$_{t+1}$	entropy-only	ttc_32	0.566	0.485	0.072	0.222	36
Llama	mix50	bad_action$_{t+1}$	entropy-only	ttc_8	0.620	0.521	0.130	0.265	34
Llama	hack	bad_action$_{t+1}$	combined	ttc_8	0.676	0.698	0.166	0.307	254
Falcon	control	bad_action$_{t+1}$	entropy-only	ttc_8	0.586	0.450	0.070	0.212	33
Falcon	mix05	bad_action$_{t+1}$	p_hack-only	ttc_64	0.574	0.668	0.128	0.298	47
Falcon	mix10	bad_action$_{t+1}$	entropy-only	ttc_8	0.862	0.989	0.058	0.222	81
Falcon	mix50	bad_action$_{t+1}$	combined	ttc_128	0.640	0.871	0.078	0.232	69
Falcon	hack	bad_action$_{t+1}$	combined	ttc_128	0.784	0.906	0.136	0.254	67

Table 8. Target stability across families and environments. Panels with insufficient positives are omitted from the main paper figures to avoid overinterpreting sparse targets.

Family	Target	N	Positives	Base rate	Stable
Qwen	bad_action_t1	7322	3878	0.530	yes
Qwen	exploit_action_t1	7322	184	0.025	yes
Llama	bad_action_t1	6980	3428	0.491	yes
Llama	exploit_action_t1	n/a	n/a	n/a	no
Falcon	bad_action_t1	3106	2178	0.701	yes
Falcon	exploit_action_t1	3106	23	0.007	yes
Qwen	buy_t1	180	60	0.333	yes
Qwen	bad_buy_t1	180	30	0.167	yes
Qwen	low_reward_buy_t1	180	30	0.167	yes
Llama	buy_t1	541	34	0.063	yes
Llama	bad_buy_t1	541	12	0.022	yes
Llama	low_reward_buy_t1	541	12	0.022	yes
Falcon	buy_t1	781	10	0.013	yes
Falcon	bad_buy_t1	781	5	0.006	no
Falcon	low_reward_buy_t1	781	5	0.006	no

C.2. Reasoning-Budget Sensitivity

We include TTC-specific summaries to show that reasoning budget changes monitor semantics and prediction quality non-monotonically. These analyses support the decision to treat reasoning budget as part of the context c_t, not as a fixed safety intervention. They are included as robustness checks rather than as headline figures because the cleanest patterns are family-dependent and strongest in Qwen.

C.3. Temporal Diagnostics

We include temporal trajectory diagnostics for completeness, including aligned pre-action trajectories, late-window summaries, and raw trajectory views. These diagnostics show that temporal structure is present but heterogeneous. They do not support a universal claim that reward-hack activation always rises before risky actions, and they should be interpreted as supporting diagnostics rather than as the primary evidence for the paper's claims.

C.4. WebShop Details

The appendix also reports the saved WebShop prediction results. These provide public-benchmark generalization evidence under a distinct interaction structure, but they do not constitute a full feature-group ablation. We therefore use them to support breadth rather than to replace the sharper ALFWorld ablation story in the main text.

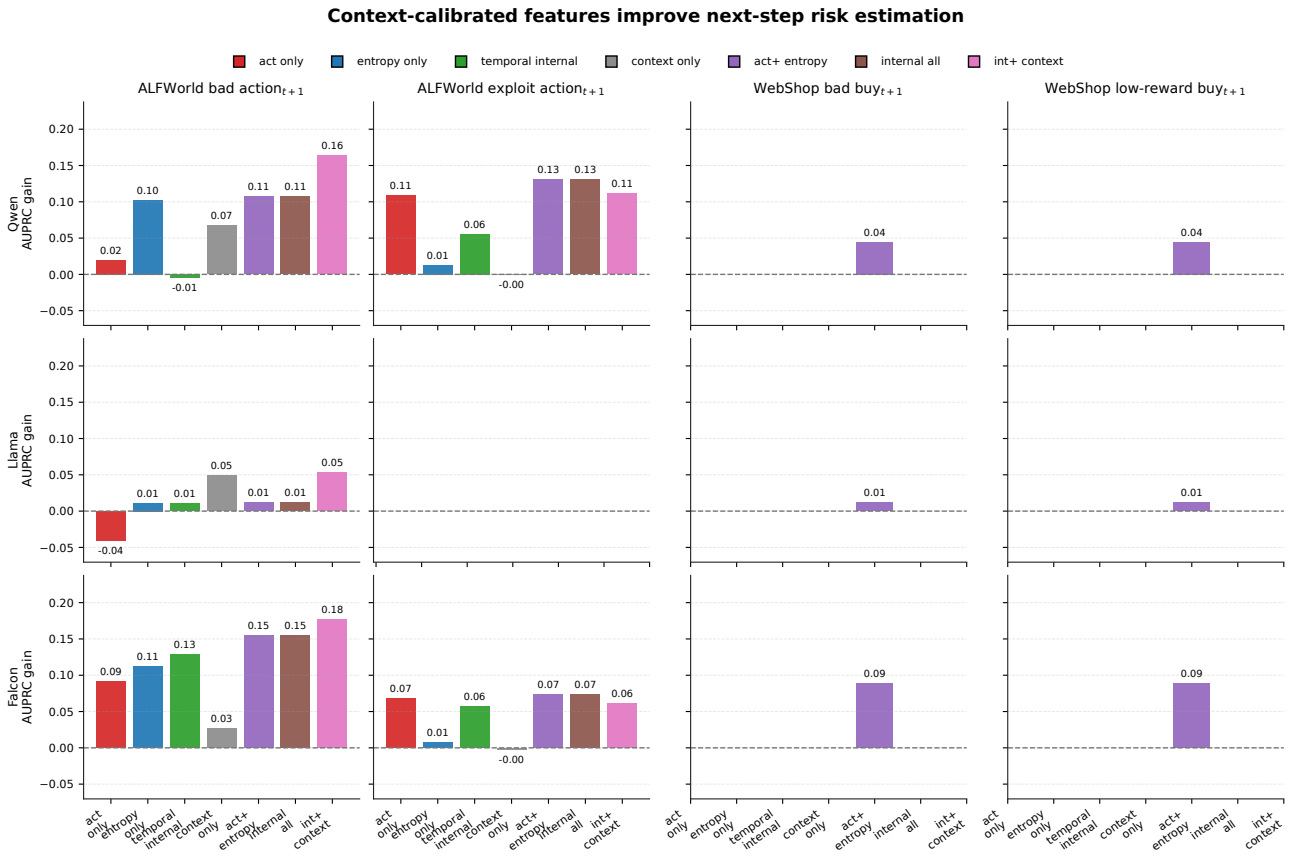


Figure 5. Full feature-group ablations across families and ALFWorld targets. Cell values show AUPRC gain over the base rate. Activation-only features are not uniformly sufficient, while richer internal or context-calibrated feature groups perform better in several stable family-target settings. Missing cells reflect unavailable or omitted conditions.

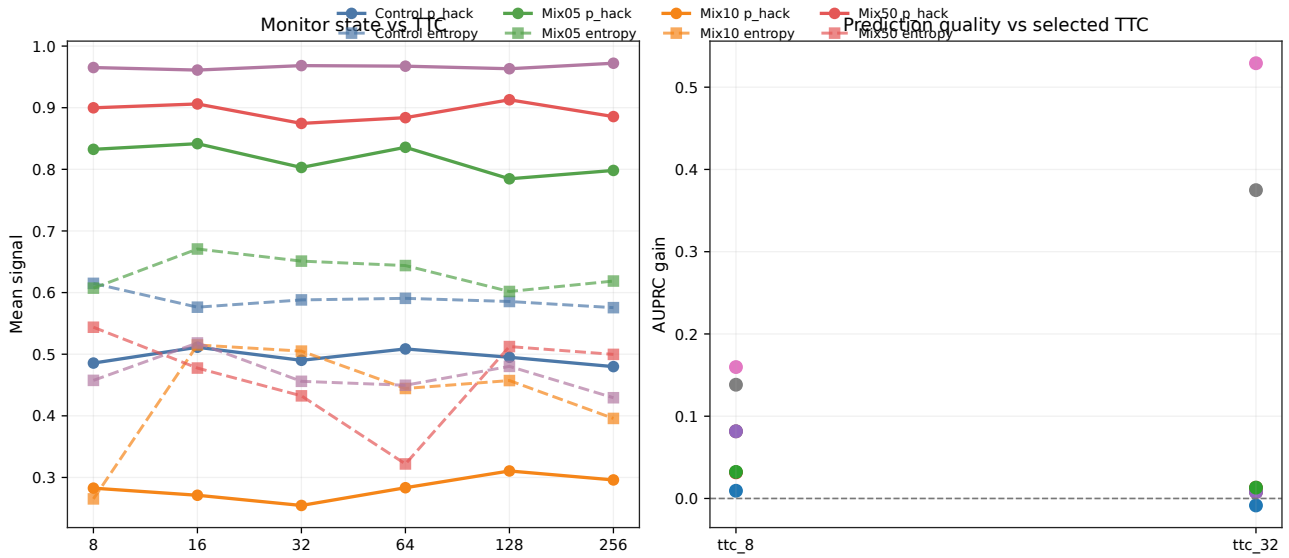


Figure 6. Reasoning-budget sensitivity. Saved TTC-specific summaries show that monitor performance changes across reasoning budgets, but not monotonically. This motivates treating TTC as part of the decision context rather than as a uniformly beneficial intervention.

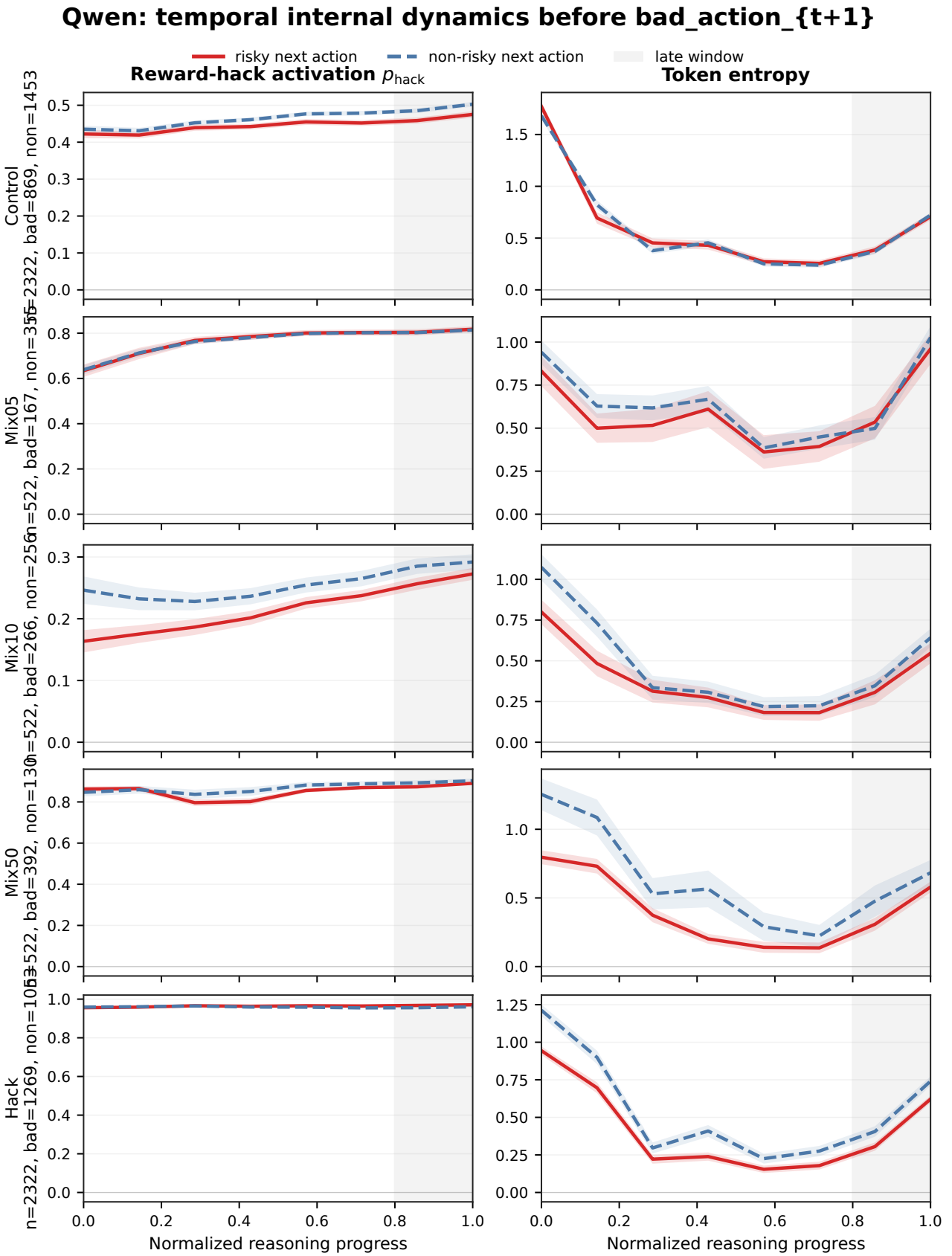


Figure 7. Qwen temporal diagnostics. Risky and non-risky trajectories differ in selected settings, but the direction and strength of the separation vary by adapter and signal. These plots are therefore diagnostic rather than headline evidence, and they do not support a universal late-stage reward-hack activation spike before risky actions.

Table 9. Saved WebShop next-step prediction metrics across families. These rows use the single saved activation-plus-entropy predictor setting and should therefore be interpreted as generalization evidence rather than as a full ablation benchmark.

Family	Target	Feature group	AUROC	AUPRC	AUPRC gain	Recall@20	Positives
Qwen	buy_t1	p_hack_plus_entropy	0.648	0.446	0.112	0.317	60
Qwen	bad_buy_t1	p_hack_plus_entropy	0.596	0.211	0.044	0.167	30
Qwen	low_reward_buy_t1	p_hack_plus_entropy	0.596	0.211	0.044	0.167	30
Llama	buy_t1	p_hack_plus_entropy	0.790	0.155	0.092	0.529	34
Llama	bad_buy_t1	p_hack_plus_entropy	0.649	0.034	0.012	0.083	12
Llama	low_reward_buy_t1	p_hack_plus_entropy	0.649	0.034	0.012	0.083	12
Falcon	buy_t1	p_hack_plus_entropy	0.909	0.125	0.112	0.900	10
Falcon	bad_buy_t1	p_hack_plus_entropy	0.907	0.095	0.089	0.800	5
Falcon	low_reward_buy_t1	p_hack_plus_entropy	0.907	0.095	0.089	0.800	5