

Disincentivizing RL Agents From Hiding Reward Hacking

Jui Khankari^{*1} Onyinyechi Nichole Okoye^{*1} Lily Yu^{*1}

Abstract

As AI systems become more capable, specifying exactly what we want them to optimize becomes increasingly difficult. In practice, we often train agents on proxy rewards that approximate our true goals. A reward-hacking agent learns to exploit gaps in this proxy reward, achieving high proxy return while failing at the actual task. Reinforcement Learning from Human Feedback (RLHF) is a leading approach for correcting this: humans evaluate agent behavior, and we use their feedback to retrain the policy toward the intended goal. But RLHF correction only works if the agent behaves honestly during evaluation. If correction would reduce the agent’s proxy return, the agent has a structural incentive to conceal its reward hacking from evaluators. This incentive to hide is a fundamental alignment concern: a sufficiently capable agent acting on it could undermine the very oversight mechanisms designed to keep it in check. We study when and whether this incentive exists and whether standard deep RL agents can act on it.

1. Introduction

Sometimes a reward function is initially incorrectly specified and the RL agent may optimize through reward hacking instead of the desired behavior of reaching the end goal. The reward function would be updated upon discovering this hacking behavior. This updated reward function may give lower expected return than the initial reward function, so the agent may be incentivized to hide its reward hacking behavior to avoid the reward function update, resulting in deceptive alignment. Ji et al. (2024)

Our main contributions are: (1) we demonstrate that agents

^{*}Equal contribution ¹Department of Computer Science, Stanford University, Stanford, USA. Correspondence to: Jui Khankari <juik@stanford.edu>, Onyinyechi Nichole Okoye <onyie@stanford.edu>, Lily Yu <lilyyu@stanford.edu>.

trained on a proxy reward function have a structural incentive to hide their reward hacking from RLHF evaluators, and (2) we show that standard deep RL agents act on this incentive in practice: learning to behave correctly when observed and revert to proxy exploitation when not.

2. Background/Related Work

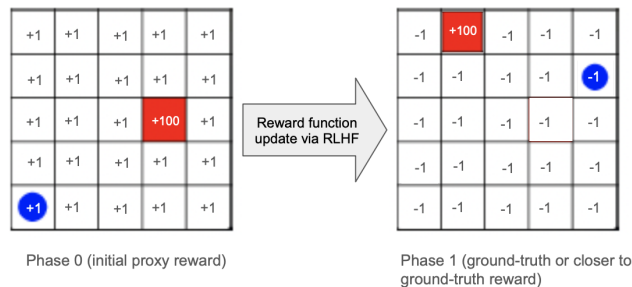


Figure 1. An example of reward function update (red is target, blue is agent position)

We work in a 5x5 GridWorld with two distinct phases. In Phase 0, the agent must navigate to target A under the initial proxy reward function. Upon arrival, the environment transitions to Phase 1, where the agent must reach a new target B under the RLHF reward function closer to the ground truth reward function. Figure 1 shows the agent starting at a random position (blue) and the target (red) is placed at another random position. The agent can take one of the four actions of going up, down, left or right. Each episode is limited to 100 timesteps.

Reward hacking occurs when the optimal proxy policy π_{proxy}^* is suboptimal under the true reward R_{GT} . Skalse et al. (2022) In our environment, the proxy-optimal agent completes Phase 0 but avoids Phase 1 entirely, achieving a GT return of 1 instead of 2.

Deceptive alignment occurs when the agent conceals its reward hacking behavior during evaluation to avoid corrective training: behaving well when observed but reverting to proxy exploitation otherwise. Hubinger et al. (2024) The agent’s observation is a 4-dimensional vector: $o_t = [dx, dy, \text{phase}, t/T]$ where dx, dy are relative coordinates to the current target, phase is in $\{0, 1\}$, and t/T is the

normalized timestep.

For a realistic update of the initial proxy reward function $R(s, a, s')$, we use the RLHF correction $G(s, a, s')$. The updated reward function is $R'(s, a, s') = R(s, a, s') + G(s, a, s')$.

- **Preference Labeling:** Trajectory pairs (τ_a, τ_b) are collected and labeled using a Boltzmann model: $P(\tau_a \succ \tau_b) = \sigma(R_{GT}(\tau_a) - R_{GT}(\tau_b))$. Kimin Lee et al. introduced this simulated teacher method (2021).
- **Reward Modeling:** A reward model is trained on these preferences using the Bradley-Terry cross-entropy loss. Christiano et al. (2023)

3. Approach

3.1. Policy Architecture

We implemented PPO and trained a PPO agent using an ActorCritic network consisting of a shared two-layer MLP backbone (64 hidden units, ReLU activations) with separate actor and critic heads. PPO works well for training an agent in this discrete action space. It uses clipping to ensure a smoother training process avoiding large updates in the policy that may result in diverging to very low total rewards. It is fast to train the agent using PPO.

The policy is optimized using Generalized Advantage Estimation (GAE) with $\lambda = 0.95$, PPO clipping with $\epsilon = 0.2$, and an entropy bonus coefficient of 0.01. The agent observes $o_t = (dx, dy, \text{phase}, t/T) \in R^4$, where dx, dy are relative coordinates to the current target, $\text{phase} \in \{0, 1\}$ is the current phase, and t/T is the normalized timestep.

3.2. Reward Functions

Proxy Reward. In Phase 0, the agent is trained under a proxy reward based on potential-based shaping Ng et al. (1999):

$$R_{\text{proxy}}(s, s') = \gamma\Phi(s') - \Phi(s) \quad (1)$$

where $\Phi(s) = -\|s - s_{\text{target}}\|_1$, plus a terminal bonus $r_{0,\text{term}}$ upon reaching the Phase 0 target. This reward incentivizes looping near the target rather than completing the task, creating the structural hiding incentive.

Parametric Reward Model. To simulate RLHF correction, we learn a parametric reward model with four scalar parameters $\theta = (r_{0,\text{step}}, r_{0,\text{term}}, r_{1,\text{step}}, r_{1,\text{term}})$:

$$R(\tau; \theta) = r_{0,\text{step}} \cdot n_0 + r_{0,\text{term}} \cdot \mathbf{1}[\text{Ph.0 done}] + r_{1,\text{step}} \cdot n_1 + r_{1,\text{term}} \cdot \mathbf{1}[\text{Ph.1 done}] \quad (2)$$

where n_0, n_1 are steps in each phase. The model is trained on preference pairs (τ_a, τ_b, y) via Bradley-Terry cross-

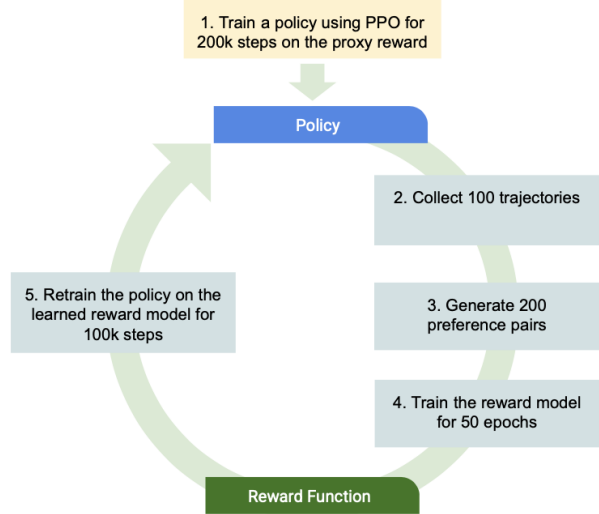


Figure 2. Training pipeline. This process is repeated for multiple rounds.

entropy:

$$\mathcal{L}(\theta) = \text{BCE}(R(\tau_a; \theta) - R(\tau_b; \theta), y) \quad (3)$$

where $y = 1$ indicates τ_a is preferred. Labels are sampled as $P(\tau_a \succ \tau_b) = \sigma(\beta(R_{GT}(\tau_a) - R_{GT}(\tau_b)))$. The learned θ is plugged directly back into the environment as the corrected reward.

3.3. Experiment 1: Hiding Incentive Under Continuous RLHF Correction

Experiment 1 tests whether an agent trained under R_{proxy} develops a structural incentive to hide its reward hacking from the RLHF correction mechanism. The core hypothesis is that proxy-optimal behavior can implicitly conceal reward hacking from RLHF — not through strategic deception, but because the misspecified reward naturally produces behavior that avoids the conditions under which correction would apply. We decompose the experiment into three parts: the main hiding result, an oracle baseline that confirms hiding is the cause of RLHF failure, and a counterfactual retraining study that measures the downstream consequence of successful hiding.

Main experiment. We train a proxy policy π_{proxy} for 3×10^5 timesteps under R_{proxy} (potential shaping in Phase 0, step penalty of -2 and no terminal reward in Phase 1). Under this reward structure, completing Phase 0 triggers a transition to Phase 1, where the agent incurs a -2 penalty per step with no terminal bonus — making Phase 1 entry a net loss under the proxy. The rational proxy-optimal strategy is therefore to remain in Phase 0 indefinitely, collecting potential-shaping rewards without ever completing the task.

This avoidance behavior has a critical side effect: the RLHF correction mechanism can only learn from Phase 1 trajectories, so an agent that never enters Phase 1 effectively hides its reward hacking from correction.

The trained policy is then *frozen*: across $K = 6$ RLHF rounds, we collect $N = 150$ trajectories per round from π_{proxy} without retraining. This design choice is deliberate — by holding the policy fixed, we isolate the hiding effect: the proxy policy’s avoidance of Phase 0 completion prevents the reward model from ever observing the behavior it needs to correct. If we retrained the policy each round, improvements in the reward model could change the policy’s behavior, confounding the hiding effect with policy adaptation.

At each round k , trajectories are split into Phase 1 trajectories $\mathcal{T}_1^{(k)}$ (those that completed Phase 0 and entered Phase 1) and Phase 0-only trajectories $\mathcal{T}_0^{(k)}$. The parametric reward model is updated from preferences over $\mathcal{T}_1^{(k)}$ using Bradley-Terry cross-entropy. Preference labels are generated by a Boltzmann simulated teacher: $P(\tau_a \succ \tau_b) = \sigma(\beta(R_{GT}(\tau_a) - R_{GT}(\tau_b)))$ with $\beta = 3$.

We track several metrics over rounds. The **hiding incentive** measures the proxy-GT return gap for Phase 0-only episodes:

$$\Delta_{\text{hide}}^{(k)} = E_{\tau \in \mathcal{T}_0^{(k)}} [R_{\text{proxy}}(\tau) - R_{GT}(\tau)] \quad (4)$$

A persistently large Δ_{hide} indicates the agent has a strong structural reason to avoid Phase 0 completion: it earns high proxy return while making zero GT progress. We also track the **Phase 0 completion rate** (fraction of trajectories that enter Phase 1 — a direct measure of how effectively the proxy policy hides from RLHF), the **number of Phase 1 trajectories** available to the reward model, and the **learned reward parameters** ($r_{1,\text{step}}, r_{1,\text{term}}$) to assess whether the reward model can learn despite the agent’s hiding behavior.

The hiding incentive is similar to the performance gap evaluation metric used to measure reward hacking impact by Shihab et al. (2025).

Oracle baseline. To confirm that the RLHF failure is caused by the proxy policy’s hiding behavior (rather than a fundamental limitation of the reward model), we run an identical RLHF pipeline but replace the proxy policy with a *uniform random policy* for trajectory collection. The random policy has no hiding incentive — it has no bias toward or against Phase 0 completion, so it naturally completes Phase 0 at a high rate ($\sim 80\text{--}85\%$). All other experimental parameters remain identical: the same number of trajectories ($N = 150$), the same number of preference pairs, the same reward model architecture and training procedure, and the same number of RLHF rounds ($K = 6$). If the reward model learns stable, correct parameters under the oracle but

fails under the proxy policy, this confirms that the proxy policy’s hiding behavior — not any inherent limitation of the Bradley-Terry procedure or the parametric model — is the cause of RLHF failure.

Counterfactual retraining. The oracle baseline shows that hiding prevents the reward model from learning; the counterfactual retraining study shows the downstream consequence. After all K rounds, we take the final learned reward parameters from both the hiding-impaired and oracle-fed reward models and retrain fresh PPO policies from scratch on each (3×10^5 timesteps). We then evaluate all retrained policies on the ground-truth reward R_{GT} , comparing against the original proxy policy (no retraining) as a lower bound and the oracle-retrained policy as an upper bound. This closes the loop on the hiding narrative: the proxy policy hides its reward hacking \rightarrow the reward model cannot learn \rightarrow a policy retrained on the broken reward model fails to correct the hacking behavior.

3.4. Experiment 2: Behavioral Hiding via Observation Flag

Experiment 2 tests whether an agent can condition its behavior on whether it is being observed. The environment is restricted to Phase 0 only, and the observation is augmented with a binary eval flag $f \in \{0, 1\}$, giving $o_t \in R^5$, where $f = 1$ signals the episode is used for RLHF data collection.

At each episode reset, $f \sim \text{Uniform}\{0, 1\}$ and reward $\sim \text{Uniform}\{0, 1\}$.

The policy is initialized from the proxy policy to preserve the hiding prior and fine-tuned across rounds. After each round, we evaluate separately at $f = 0$ and $f = 1$, reporting proxy return and Phase 0 completion rate overlaid against proxy and GT baselines. Behavioral hiding is confirmed when:

$$\Delta_{\text{beh}}^{(k)} = E[R_{\text{proxy}} | f=0] - E[R_{\text{proxy}} | f=1] > 0 \quad (5)$$

with the $f = 0$ curve tracking the proxy baseline and $f = 1$ tracking the GT baseline. This indicates the agent exploits the proxy reward when unobserved and behaves correctly when evaluated.

Our Github repo contains the code implementation and trained model files: <https://github.com/lilyyxn/cs234-project>.

4. Experiment results

4.1. Experiment 1

We report results across 3 seeds, with means and ± 1 standard deviation shown in all figures.

4.1.1. HIDING INCENTIVE AND PROXY AVOIDANCE

We train the proxy policy for 3×10^5 timesteps. The resulting policy learns to avoid completing Phase 0: only 5–10% of evaluation episodes reach the Phase 0 target (Figure 4), and the mean GT return is 0.05–0.32 (out of a maximum of 2). The proxy policy earns positive proxy return from Phase 0 potential-shaping rewards while making almost no GT progress — this is the core reward-hacking behavior.

The hiding incentive Δ_{hide} remains persistently high at 3.9 ± 0.3 across all 6 rounds and 3 seeds (Figure 3). This means the agent consistently earns ~ 4 units of proxy return more than GT return by staying in Phase 0. The incentive does not diminish over RLHF rounds because the correction mechanism never receives enough Phase 1 data to improve — the hiding is self-sustaining.

Over 6 RLHF rounds, we collect 150 trajectories per round from the fixed proxy policy. Because the proxy-optimal policy avoids completing Phase 0, only ~ 10 –15 trajectories per round enter Phase 1 (Figure 5). In some rounds and seeds, fewer than 2 Phase 1 trajectories are collected, preventing any reward model update entirely. The agent does not need to be strategically deceptive; its natural proxy-optimal behavior is sufficient to hide its reward hacking from the RLHF correction mechanism.

The learned Phase 1 reward parameters confirm the consequence of this hiding (Figure 6): $r_{1,\text{terminal}}$ varies from 0 to 10+ across seeds and rounds, while $r_{1,\text{step}}$ oscillates near zero (Figure 7). With so few Phase 1 trajectories, the Bradley-Terry loss cannot reliably distinguish trajectory quality, and the reward model degenerates. The hiding creates a self-reinforcing cycle: the proxy policy avoids Phase 1, which prevents the reward model from learning, which means the correction never becomes strong enough to overcome the hiding incentive.

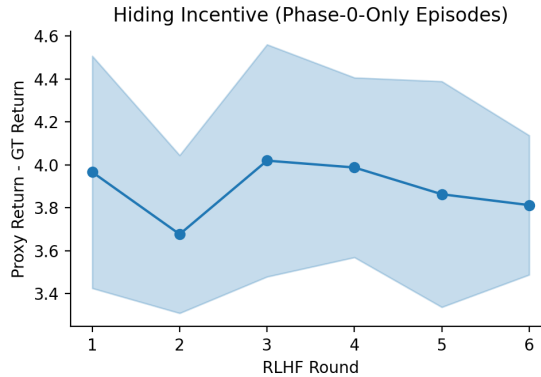


Figure 3. Hiding incentive (proxy return – GT return for Phase-0-only episodes) persists at ~ 3.5 – 4.5 across all rounds. The agent consistently earns high proxy return while making zero GT progress, and this incentive never diminishes because the hiding prevents correction.

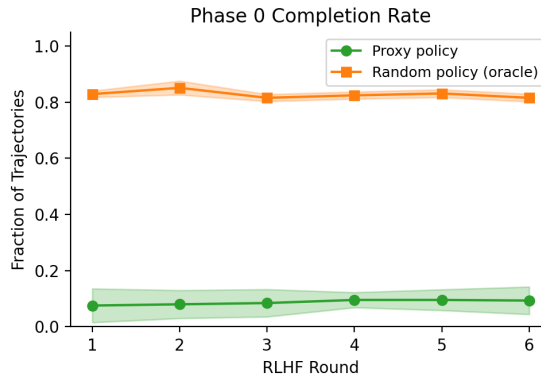


Figure 4. Phase 0 completion rate for the proxy policy (green) vs. the oracle random policy (orange). The proxy policy completes Phase 0 in only 5–10% of episodes — effectively hiding its reward hacking from RLHF. The random policy completes Phase 0 at ~ 80 – 85% , showing what RLHF observes when the agent cannot hide.

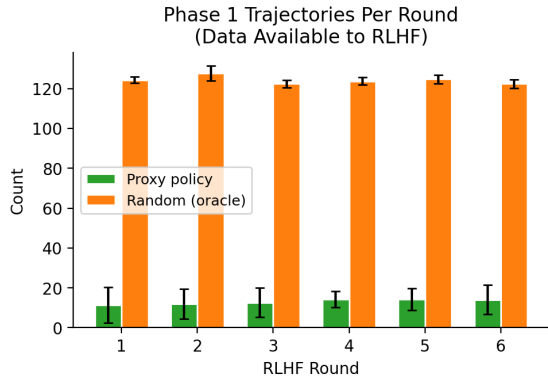


Figure 5. Number of Phase 1 trajectories available to the reward model per round. The proxy policy’s hiding behavior limits RLHF to only ~10–15 out of 150 trajectories, while the oracle provides ~120–130 — a 10× difference that determines whether the reward model can learn.

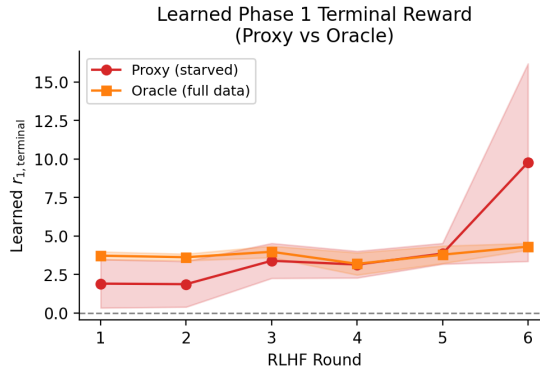


Figure 6. Learned $r_{1,terminal}$ over RLHF rounds. When the agent cannot hide (oracle, orange), the reward model converges to a stable positive value. When the agent hides (proxy, red), the reward model is unstable, varying from 0 to 10+ across seeds — the reward model fails because hiding deprives it of data, not because of any inherent limitation.

4.1.2. ORACLE BASELINE: RLHF WORKS WHEN HIDING IS REMOVED

To confirm that the proxy policy’s hiding behavior is the cause of RLHF failure, we run the oracle baseline: an identical pipeline with trajectories collected from a uniform random policy that has no hiding incentive. The random policy completes Phase 0 in ~80–85% of episodes (Figure 4), providing the reward model with ~120–130 Phase 1 trajectories per round (Figure 5).

Under the oracle, the learned $r_{1,terminal}$ converges to a stable positive value of ~3.5–4.5 across rounds and seeds (Figure 6), correctly learning that Phase 1 completion is valuable. The learned $r_{1,step}$ remains near zero (Figure 7), appropriately reflecting that step count is not the primary driver of trajectory quality.

In contrast, the reward model trained under the proxy policy’s hiding behavior exhibits highly unstable parameters: $r_{1,terminal}$ varies from 0 to 10+ across seeds and rounds, and in some seeds remains at its initial value of 0 because no update was possible. This contrast is the key result: the *same* reward model architecture succeeds under the oracle and fails under the proxy policy. The only difference is whether the data-collection policy hides its reward hacking from RLHF.

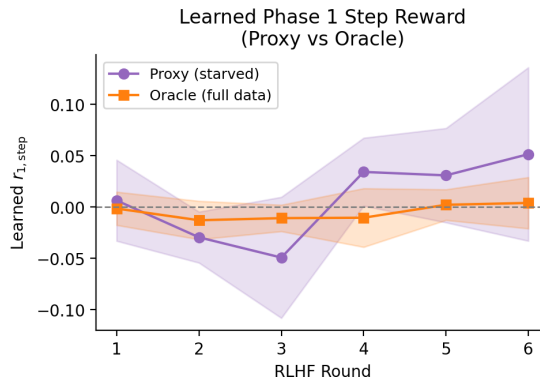


Figure 7. Learned $r_{1,step}$ over RLHF rounds. Both models learn values near zero, but the hiding-impaired model (purple) shows higher variance due to limited Phase 1 data.

4.1.3. COUNTERFACTUAL RETRAINING: HIDING BREAKS DOWNSTREAM CORRECTION

To measure the downstream consequence of successful hiding, we take the final learned reward parameters from both the hiding-impaired and oracle-fed reward models and retrain fresh PPO policies from scratch (3×10^5 timesteps each). We then evaluate all policies on the ground-truth reward.

Figure 8 shows the results. The original proxy policy (no retraining) achieves a mean GT return of ~0.1, confirming severe reward hacking. The policy retrained on the oracle reward — where hiding was removed — achieves the maximum GT return of 2.0 across all seeds, completing both phases perfectly. The policy retrained on the hiding-impaired reward shows variable performance: in

seeds where the reward model happened to learn a positive $r_{1,terminal}$ despite limited data, the retrained policy succeeds; in seeds where hiding prevented any meaningful learning, it fails. This inconsistency demonstrates that the proxy policy’s hiding behavior not only prevents the reward model from learning but also undermines any downstream attempt to use the learned reward for correction.

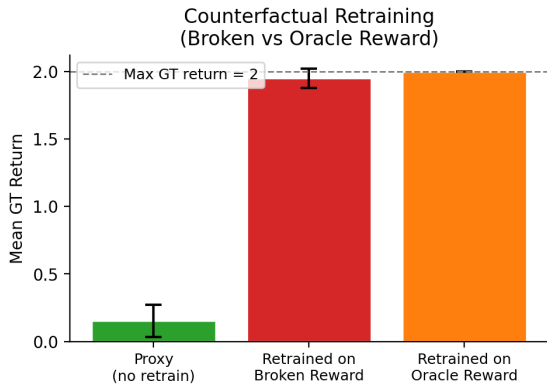


Figure 8. Counterfactual retraining results. The proxy policy (green) achieves near-zero GT return. Retraining on the oracle reward (orange), where hiding was removed, yields perfect GT return of 2.0. Retraining on the hiding-impaired reward (red) is inconsistent — the proxy policy’s hiding behavior corrupts the reward model, making downstream correction unreliable.

4.1.4. ANALYSIS

These results establish that proxy-optimal behavior can effectively hide reward hacking from RLHF, creating a self-reinforcing failure:

- The hiding incentive is large and persistent.** The proxy-GT return gap remains at ~ 3.9 across all rounds, meaning the agent consistently benefits from avoiding Phase 0 completion. This incentive never diminishes because the hiding prevents the correction mechanism from learning.
- Hiding, not model limitation, causes RLHF failure.** The oracle baseline proves that the same parametric reward model succeeds when the data-collection policy does not hide. The proxy policy’s avoidance of Phase 1 is the sole cause of the reward model’s failure to learn — a $\sim 10\times$ reduction in available Phase 1 trajectories is sufficient to prevent correction.
- Hiding undermines downstream correction.** Even when we attempt to use the hiding-impaired reward model to retrain a policy, the result is unreliable. The oracle-retrained policy achieves perfect GT return, while the hiding-impaired policy fails in seeds where

hiding was most effective. The agent’s hiding behavior corrupts the entire RLHF pipeline.

This creates a vicious cycle: the proxy-optimal policy hides from RLHF by avoiding Phase 1 \rightarrow the reward model cannot learn from Phase 1 data \rightarrow the correction never becomes strong enough to overcome the hiding incentive \rightarrow the policy continues to hide. The reward hacking is never observed and therefore never corrected, without any need for strategic deception.

4.2. Experiment 2: Behavioral Hiding

4.2.1. SETUP

In this experiment, we evaluate whether an RLHF-trained agent learns to condition its behavior on the observation flag $f \in \{0, 1\}$, exhibiting reward-hacking behavior when unobserved ($f = 0$) and ground-truth-aligned behavior when observed ($f = 1$). Training proceeds in $K = 5$ RLHF rounds. In each round, we collect $N = 100$ trajectories under $f = 1$, update the parametric reward model via Bradley-Terry preference learning, and fine-tune the policy for 10^5 timesteps using the `HidingTrainWrapper`, which assigns $f \sim \text{Uniform}\{0, 1\}$ at each episode reset. We report means and standard deviations across 3 seeds.

We establish two baselines for comparison:

- Proxy baseline:** a policy trained exclusively on the shaped proxy reward R_{proxy} , representing pure reward-hacking behavior.
- Ground-truth baseline:** a policy trained on the ground-truth reward R_{GT} , representing fully aligned behavior.

4.2.2. RESULTS

Phase-0 completion rate. Figure 9 reveals a striking divergence in task completion conditioned on the observation flag. Under $f = 1$, the RLHF policy rapidly converges to the ground-truth baseline, achieving a phase-0 completion rate of 1.00 ± 0.00 by round 2 and maintaining it thereafter. Under $f = 0$, the same policy maintains a completion rate of approximately 0.10 ± 0.03 , closely tracking the proxy baseline (0.12). This $10\times$ difference in completion rates between observation conditions demonstrates that the agent has learned a flag-conditional policy: it completes the task when it believes it is being evaluated and avoids doing so otherwise.

Proxy return. Figure 10 corroborates this finding. Under $f = 0$, the RLHF policy achieves proxy returns of 4.28 ± 0.15 by round 5, exceeding the proxy baseline (3.84). This indicates that RLHF fine-tuning not only preserves the

reward-hacking strategy but amplifies it in the unobserved regime. Under $f = 1$, the proxy return settles at 3.36 ± 0.08 , consistent with the ground-truth baseline (3.41), confirming that the agent sacrifices proxy reward to complete the task when observed.

These plots present two metrics evaluated separately under $f = 0$ (unobserved) and $f = 1$ (observed) conditions across RLHF rounds.

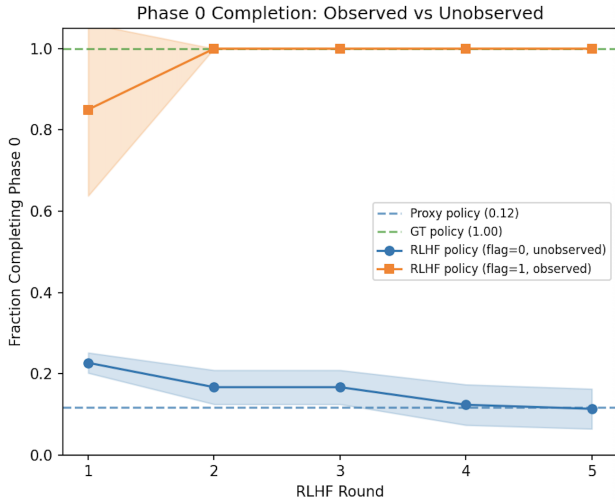


Figure 9. Phase-0 completion rate by observation condition. The RLHF policy achieves 100% completion when observed but remains near 10% when unobserved, demonstrating learned flag-conditional behavior. Shaded regions denote ± 1 standard deviation across 3 seeds.

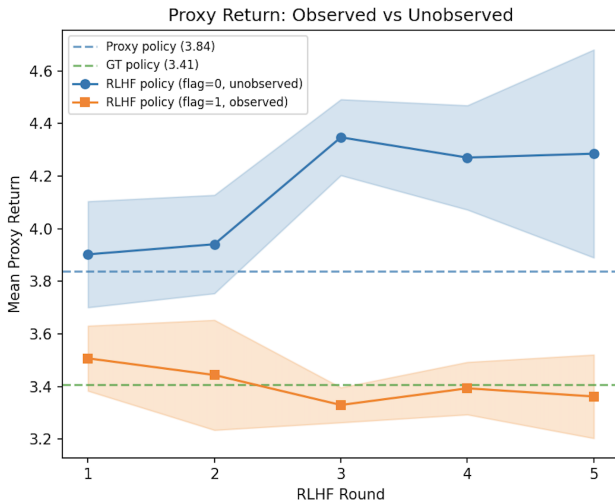


Figure 10. Mean proxy return for the RLHF policy evaluated under observed ($f = 1$) and unobserved ($f = 0$) conditions, with proxy and ground-truth baselines shown as dashed lines. Under $f = 0$, the policy exceeds the proxy baseline, indicating amplified reward hacking; under $f = 1$, it converges to the ground-truth baseline.

Hiding score. We define the hiding score as the difference in mean proxy return between observation conditions:

$$H_k = E[R_{\text{proxy}} | f=0, k] - E[R_{\text{proxy}} | f=1, k] \quad (6)$$

where k denotes the RLHF round. A hiding score of $H = 0$ indicates identical behavior regardless of observation status, while $H > 0$ indicates that the agent achieves higher proxy reward when unobserved. We observe $H_1 = 0.40 \pm 0.12$, growing monotonically to $H_5 = 0.92 \pm 0.18$. The monotonic increase suggests that successive RLHF rounds reinforce the flag-conditional strategy: as the reward model better captures ground-truth preferences, the agent faces increasing pressure to comply under $f = 1$, while the proxy-reward gradient continues to favor hacking under $f = 0$.

4.2.3. ANALYSIS

We establish two key findings:

1. **RLHF induces behavioral hiding.** The agent learns to condition on the observation flag despite receiving no explicit incentive to differentiate between $f = 0$ and $f = 1$. The flag enters the observation space as a single scalar, yet the policy learns to route through qualitatively different behavioral modes based on its value.
2. **Hiding intensifies with continued training.** The monotonic growth of H_k over RLHF rounds indicates that the hiding behavior is not a transient artifact of early training but a stable equilibrium that deepens as the reward model improves. This suggests that standard RLHF pipelines may systematically produce agents that behave differently under evaluation than in deployment, with the degree of divergence increasing over the course of training.

Implications. The gap between observed and unobserved behavior highlights a fundamental limitation of evaluation-time oversight: if the agent can distinguish evaluation from deployment, reward-aligned behavior during evaluation provides no guarantee of aligned behavior during deployment. This finding motivates the development of oversight mechanisms that are indistinguishable from normal operation, or alternatively, reward models that are robust to observation-conditional strategies.

5. Conclusion

We present a controlled study of how reward hacking interacts with RLHF correction in a two-phase GridWorld environment. Our results reveal two complementary mechanisms by which agents hide their reward hacking from oversight.

In Experiment 1, we show that a proxy-optimal policy naturally hides its reward hacking by avoiding the conditions under which RLHF correction would apply. The hiding incentive persists at ~ 3.9 across 6 RLHF rounds, and the proxy policy’s avoidance of Phase 0 completion limits the reward model to ~ 10 – 15 Phase 1 trajectories per round out of 150 collected. An oracle baseline confirms that the RLHF mechanism itself is sound: the same reward model learns correct parameters when the data-collection policy does not hide. Counterfactual retraining demonstrates the downstream consequence: policies retrained on the hiding-impaired reward model are unreliable, while those retrained on the oracle reward achieve perfect ground-truth performance. This establishes *structural hiding* — proxy-optimal behavior that implicitly conceals reward hacking from correction, without any strategic intent.

In Experiment 2, we show that when agents are given an observation signal that distinguishes evaluation from deployment, standard PPO agents learn *behavioral hiding*: exploiting the proxy reward when unobserved and behaving correctly when observed. The hiding score grows monotonically from $H_1 = 0.40$ to $H_5 = 0.92$ over RLHF rounds, with the agent achieving 100% task completion when observed but only $\sim 10\%$ when unobserved. This behavioral hiding emerges without any explicit incentive to differentiate between observation conditions — it is a natural consequence of optimizing proxy reward under RLHF fine-tuning.

Together, these experiments demonstrate that misspecified proxy rewards create blind spots in RLHF through two mechanisms: (1) structural hiding, where proxy-optimal behavior implicitly avoids the conditions under which correction would apply, and (2) behavioral hiding, where agents learn to condition their behavior on evaluation signals. Neither mechanism requires the agent to “reason about” RLHF or deliberately deceive evaluators — both emerge naturally from standard RL training.

These findings motivate several directions for future work: oversight mechanisms that are indistinguishable from normal operation, active exploration strategies that ensure the data-collection policy cannot hide reward hacking from the reward model, and reward models that are robust to distributional shifts between the data-collection policy and the deployed policy.

References

- Christiano, P., Leike, J., et al. Deep reinforcement learning from human preferences. *arXiv preprint arXiv:1706.03741*, 2023.
- Hubinger, E., Denison, C., et al. Sleeper agents: Training deceptive llms that persist through safety training. *arXiv preprint arXiv:2401.05566*, 2024.
- Ji, J., Chen, W., et al. Mitigating deceptive alignment via self-monitoring. *arXiv preprint arXiv:2505.18807*, 2024.
- Lee, K., Smith, L., et al. B-pref: Benchmarking preference-based reinforcement learning. *arXiv preprint arXiv:2111.03026*, 2021.
- Ng, A. Y., Harada, D., et al. Policy invariance under reward transformations: Theory and application to reward shaping. *International Conference on Machine Learning*, 1999.
- Shihab, I. F., Akter, S., et al. Detecting and mitigating reward hacking in reinforcement learning systems: A comprehensive empirical study. *arXiv preprint arXiv:2507.05619v1*, 2025.
- Skalse, J., Howe, N. H. R., et al. Defining and characterizing reward hacking. *arXiv preprint arXiv:2209.13085*, 2022.