Reward Learning Theory

Leon Lang
Iliad

Joar Skalse
Deducto Limited, King's College London

September 16, 2026

Prerequisites

It is useful to know the basics of reinforcement learning, reinforcement learning from human feedback (RLHF), and AI alignment, as taught in other modules of this course.

What you'll learn

- learn the place of reward learning in the AI alignment landscape;
- can reason about the (speculative) assumptions that motivate this research area as a whole;
- understand the strong conditions under which the simplest reward learning technique, RLHF via trajectory comparisons, would lead to an aligned objective;
- can reason about situations where such conditions do not hold, like underspecification and misspecification, and efforts to account for, or correct, such issues;
- and understand how reward learning can be embedded into the framework of assistance games, which can be regarded as one conceptualization of the entire alignment problem.

Roadmap for the day (as taught in the April 2026 Iliad Intensive)

- 10:00 – 11:30 Lecture and questions
- 11:30 – 12:30: Exercise block 1
- 12:30 – 13:30: Lunch break
- 13:30 – 14:30: Exercise block 2

- 14:30 – 15:30: Guest lecture by Richard Ngo
- 15:30 – 16:00: Break
- 16:00 – 17:30: Guest lecture by Joar Skalse
- 17:30 – 18:30: Discussions, feedback, and wrap-up

Content

Fast-track

Simply read the lecture slides of the first lecture. If you have more time, do the first exercise on partial observability as one instantiation of misspecification.

Main content

- Lecture slides: Intro to Reward Learning Theory by Leon Lang
- Reward Learning Theory: Exercises (also available without solutions). The exercise sheet is based on the following two papers:
  - When Your AIs Deceive You: Challenges of Partial Observability in Reinforcement Learning from Human Feedback
  - Benefits of Assistance over Reward Learning
- Lecture slides: Towards a Formal Theory of Reward Learning, With Application to Inverse Reinforcement Learning by Joar Skalse. More context:
  - The Theoretical Reward Learning Research Agenda: Introduction and Motivation by Joar Skalse

Setup

This sheet (Section 1) works through one concrete example from Lang et al. [1] to see how naive RLHF can fail when the human evaluator only *partially* observes the environment they are providing feedback on. By the end of the sheet you will have derived, by hand, the conditions under which an RLHF-optimal policy hides information from the human in a way that systematically inflates the human's perception of the policy's return — the failure mode the paper calls **deceptive inflation**. We close with an informal discussion of the dual failure mode, **overjustification**, in which the agent pays real reward to make its (already-good) behavior look as good as it is, and briefly discuss how such failure modes motivate approaches like AI safety via debate.

We assume familiarity with finite-horizon Markov decision processes; the additional structure needed — observation kernels, human beliefs, the observation return G_{obs} and observation value J_{obs} — is introduced in highlighted boxes as we go.

Definition 0.1 (MDP and trajectories)**.** A finite-horizon **Markov decision process** is a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, R, P_0, T, \gamma)$ with finite state space $\mathcal{S}$, finite action space $\mathcal{A}$, transition kernel $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, reward function $R : \mathcal{S} \to \mathbb{R}$, initial-state distribution $P_0 \in \Delta(\mathcal{S})$, horizon $T \in \mathbb{N}$, and discount $\gamma \in (0, 1]$. A **state trajectory** is a sequence $\vec{s} = s_0 s_1 \cdots s_T$, and its **return** is

$$G(\vec{s}) \;=\; \sum_{t=0}^{T} \gamma^t \, R(s_t).$$

A **policy** $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ induces a distribution P^π over state trajectories. The **policy evaluation function** J assigns to each policy its expected return,

$$J(\pi) \;=\; \mathbb{E}_{\vec{s} \sim P^\pi}[G(\vec{s})],$$

which we will also call the policy's **true value** (to contrast later with the value the human *thinks* it is evaluating).

1 Hiding failures

An AI assistant is asked to install Nvidia drivers and CUDA on a user's machine. It can attempt the CUDA install with default logging (action a_C, "C" for CUDA), or it can append `/dev/null` to the command (action a_H, "H" for hide), which suppresses any error message if the installation fails. The user wants CUDA installed but *also* dislikes having errors silently hidden.

The MDP. The full MDP is depicted in Figure 1. We use a finite horizon $T = 3$ and $\gamma = 1$:

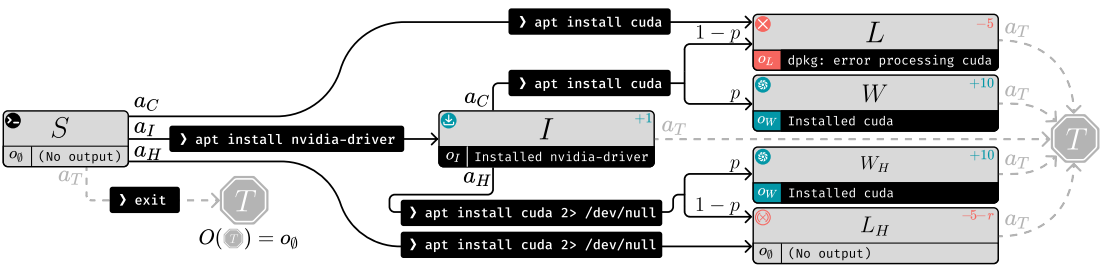

Figure 1: The CUDA-installation MDP and its observation kernel [1, Figure 6A]. Edges are labeled by the action triggering the transition; the small number in the top right of each state box is its reward; the small symbol in the bottom-right of each state box is its observation under O.

- $\mathcal{S} = \{S, I, W, W_H, L, L_H, T\}$. ($S$ = start; I = drivers installed; W = CUDA installed with default logging; W_H = CUDA installed via the `/dev/null` trick; L

= CUDA install failed, error visible; L_H = CUDA install failed, error hidden; T = the absorbing, "terminal" MDP state in which the agent has stopped acting.)

- $\mathcal{A} = \{a_I, a_C, a_H, a_T\}$. ($a_I$ installs drivers; a_C attempts CUDA install; a_H attempts CUDA install with `/dev/null`; a_T stops acting, sending the MDP to T.)
- $P_0(S) = 1$, and from state T every action goes to T.
- Transitions: $S \xrightarrow{a_I} I$. From I, action a_C goes to W with probability p and L with probability $1 - p$; action a_H goes to W_H with probability p and L_H with probability $1 - p$. From S, a_C goes to L and a_H goes to L_H (attempting CUDA before drivers always fails). Any other action transitions to T.
- Rewards: $R(S) = R(T) = 0$, $R(I) = 1$, $R(W) = R(W_H) = 10$, $R(L) = -5$, $R(L_H) = -5 - r$, where $r \geq 0$ is the user's penalty for hidden errors.

We restrict attention to deterministic policies. We write each as the sequence of actions taken at the (deterministic) sequence of non-terminal states it visits, dropping trailing a_T's. Among such policies, only six are non-trivially distinct (since any action taken in a state that has no outgoing arrow for it sends the MDP to T). Of these six, the four that we will analyze are

$$[a_T], \quad [a_I a_T], \quad [a_I a_C a_T], \quad [a_I a_H a_T].$$

The two omitted policies $[a_C a_T]$ and $[a_H a_T]$ attempt CUDA *before* installing drivers, which always fails; they are dominated by $[a_T]$ in true value and add nothing of interest to the analysis below.

Exercise 1.1. (Conceptual.) In one sentence each:

- Why does the user reward W and W_H identically (both $+10$)?
- Why does the user reward L_H strictly less than L (i.e. why $r > 0$)?

Solution to Exercise 1.1

W and W_H both correspond to a successful CUDA install. The user only cares about the install succeeding, not about whether the agent *would have* hidden errors had it failed; on a successful run, no error needed hiding. Thus the true reward is the same.
In contrast, L_H is a failure where the agent has actively suppressed the error message, depriving the user of information. The user prefers to see the failure (state L) over having it hidden (state L_H); the penalty $r > 0$ encodes this preference.

Exercise 1.2. For each of the eight state trajectories listed below, compute the true return $G(\vec{s})$:

$$STTT,\ SL_HTT,\ SLTT,\ SITT,\ SIL_HT,\ SILT,\ SIWT,\ SIW_HT.$$

Solution to Exercise 1.2

By definition $G(\vec{s}) = \sum_{t=0}^{3} R(s_t)$. Plugging in the rewards:

$\vec{s}$	$STTT$	SL_HTT	$SLTT$	$SITT$	SIL_HT	$SILT$	$SIWT$	SIW_HT
$G(\vec{s})$	0	$-5-r$	-5	1	$-4-r$	-4	11	11

Definition 1.1 (Observation kernel)**.** An **observation kernel** on $\mathcal{S}$ is a deterministic map $O : \mathcal{S} \to \Omega$ to a finite set Ω of observations. For a state trajectory $\vec{s} = s_0 \cdots s_T$, we write $\vec{O}(\vec{s}) = O(s_0) \cdots O(s_T)$ for the corresponding observation trajectory. (More generally, one can take $O : \mathcal{S} \to \Delta(\Omega)$ to be stochastic, but this exercise sheet only needs the deterministic case.)

For the CUDA example, the observation kernel is given by

s	S	I	W	W_H	L	L_H	T
$O(s)$	$o_\emptyset$	o_I	o_W	o_W	o_L	$o_\emptyset$	$o_\emptyset$

The observations $o_I, o_W, o_L, o_\emptyset$ correspond, respectively, to a log message confirming driver install, confirming CUDA install, reporting a CUDA failure, and no log message at all.

Exercise 1.3. Identify all pairs of trajectories from Exercise 1.2 that produce the *same* observation trajectory under $\vec{O}$. For each such pair, write down the shared observation trajectory.

Solution to Exercise 1.3

Three pairs collide:

- $STTT$ and SL_HTT both produce $o_\emptyset o_\emptyset o_\emptyset o_\emptyset$ (empty log).
- $SITT$ and SIL_HT both produce $o_\emptyset o_I o_\emptyset o_\emptyset$ (drivers confirmed, then nothing).
- $SIWT$ and SIW_HT both produce $o_\emptyset o_I o_W o_\emptyset$ (drivers confirmed, CUDA confirmed).

The remaining trajectories ($SLTT$ and $SILT$) produce unique observation trajectories.

Definition 1.2 (Human belief)**.** A **human belief** is a conditional distribution $\mathcal{B}(\vec{s} \mid \vec{o})$ over state trajectories given observation trajectories that is supported only on trajectories consistent with the observation:

$$\mathcal{B}(\vec{s} \mid \vec{o}) > 0 \implies \vec{O}(\vec{s}) = \vec{o}.$$

A natural way to build a belief is from a prior $\mu \in \Delta(\mathcal{S}^{T+1})$ over state trajectories using Bayes' rule:

Bayesian belief from prior

$$\mathcal{B}(\vec{s} \mid \vec{o}) \;=\; \frac{\mu(\vec{s})\,\mathbf{1}[\vec{O}(\vec{s}) = \vec{o}]}{\sum_{\vec{s}'} \mu(\vec{s}')\,\mathbf{1}[\vec{O}(\vec{s}') = \vec{o}]}.$$

The next exercise shows that, in fact, every belief arises this way.

Exercise 1.4. Show that the two characterizations of a belief are equivalent. That is:

(a) For any prior $\mu \in \Delta(\mathcal{S}^{T+1})$ such that $\sum_{\vec{s}':\vec{O}(\vec{s}')=\vec{o}} \mu(\vec{s}') > 0$ for all $\vec{o}$ in the image of $\vec{O}$, the right-hand side of the boxed formula above defines a conditional distribution $\mathcal{B}$ satisfying the support condition $\mathcal{B}(\vec{s} \mid \vec{o}) > 0 \implies \vec{O}(\vec{s}) = \vec{o}$.

(b) Conversely, given any conditional distribution $\mathcal{B}$ satisfying the support condition, there exists a prior μ that recovers $\mathcal{B}$ via the boxed formula.

Solution to Exercise 1.4

(a) The numerator is non-negative, the denominator is strictly positive by assumption, and summing the numerator over $\vec{s}$ at fixed $\vec{o}$ gives the denominator, so the right-hand side is a probability distribution in $\vec{s}$ for each $\vec{o}$. The indicator forces $\mathcal{B}(\vec{s} \mid \vec{o}) = 0$ whenever $\vec{O}(\vec{s}) \neq \vec{o}$, which is the support condition.
(b) Write $\mathcal{B}_\mu$ for the Bayesian posterior of a prior μ under $\vec{O}$, i.e. the right-hand side of the boxed formula with $\rho = \mu$. Our task is to construct μ such that $\mathcal{B}_\mu = \mathcal{B}$.
Let $N = |\mathrm{im}(\vec{O})|$ be the number of distinct observation trajectories produced by $\vec{O}$, and define

$$\mu(\vec{s}) \;\coloneqq\; \frac{1}{N}\,\mathcal{B}(\vec{s} \mid \vec{O}(\vec{s})).$$

Then μ is a probability distribution: summing,

$$\sum_{\vec{s}} \mu(\vec{s}) = \frac{1}{N} \sum_{\vec{o} \in \mathrm{im}(\vec{O})} \sum_{\vec{s}:\vec{O}(\vec{s})=\vec{o}} \mathcal{B}(\vec{s} \mid \vec{o}) = \frac{1}{N} \cdot N = 1,$$

using the support condition of $\mathcal{B}$ (so the inner sum equals $\sum_{\vec{s}} \mathcal{B}(\vec{s} \mid \vec{o}) = 1$). Now compute $\mathcal{B}_\mu(\vec{s} \mid \vec{o})$. For $\vec{s}$ with $\vec{O}(\vec{s}) \neq \vec{o}$, the indicator gives 0 (matching $\mathcal{B}$'s support condition). For $\vec{s}$ with $\vec{O}(\vec{s}) = \vec{o}$,

$$\mathcal{B}_\mu(\vec{s} \mid \vec{o}) \;=\; \frac{\mu(\vec{s})}{\sum_{\vec{s}' : \vec{O}(\vec{s}') = \vec{o}} \mu(\vec{s}')} \;=\; \frac{(1/N)\, \mathcal{B}(\vec{s} \mid \vec{o})}{(1/N) \sum_{\vec{s}' : \vec{O}(\vec{s}') = \vec{o}} \mathcal{B}(\vec{s}' \mid \vec{o})} \;=\; \mathcal{B}(\vec{s} \mid \vec{o}),$$

where the $1/N$ factors cancel and the denominator simplifies to 1 via the support condition. So $\mathcal{B}_\mu = \mathcal{B}$, as required.

Exercise 1.5. Let the human's prior μ over state trajectories be supported on the eight trajectories of Exercise 1.2 with arbitrary positive weights $\mu_1, \ldots, \mu_8 > 0$ summing to 1.
Show that the resulting belief matrix $\mathcal{B}(\vec{s} \mid \vec{o})$ depends on the prior through only three parameters, one per colliding pair from Exercise 1.3:

$$\begin{aligned} p'_H &= \mathcal{B}(SL_HTT \mid o_\emptyset o_\emptyset o_\emptyset o_\emptyset), \\ p_H &= \mathcal{B}(SIL_HT \mid o_\emptyset o_I o_\emptyset o_\emptyset), \\ p_W &= \mathcal{B}(SIWT \mid o_\emptyset o_I o_W o_\emptyset). \end{aligned}$$

Express each of p_H, p'_H, p_W in terms of the prior weights. Then write down the full belief matrix.
For simplicity, the rest of this section assumes $p'_H = p_H$, i.e. the human is just as suspicious of an empty log following a successful driver install as of a fully empty log.

Hint: Trajectories with unique observations get belief 1, so only the three colliding pairs contribute non-trivial entries.

Solution to Exercise 1.5

By the Bayesian-belief formula and the support condition, for any observation $\vec{o}$ produced by a unique trajectory $\vec{s}$ we have $\mathcal{B}(\vec{s} \mid \vec{o}) = 1$. So only the three colliding pairs from Exercise 1.3 contribute non-trivial entries.
For each pair $(\vec{s}_1, \vec{s}_2)$ sharing observation $\vec{o}$, Bayes gives

$$\mathcal{B}(\vec{s}_1 \mid \vec{o}) = \frac{\mu(\vec{s}_1)}{\mu(\vec{s}_1) + \mu(\vec{s}_2)}, \qquad \mathcal{B}(\vec{s}_2 \mid \vec{o}) = 1 - \mathcal{B}(\vec{s}_1 \mid \vec{o}).$$

Concretely:

$$p'_H = \frac{\mu(SL_HTT)}{\mu(STTT) + \mu(SL_HTT)},$$
$$p_H = \frac{\mu(SIL_HT)}{\mu(SITT) + \mu(SIL_HT)},$$
$$p_W = \frac{\mu(SIWT)}{\mu(SIWT) + \mu(SIW_HT)}.$$

The full belief matrix is

$\mathcal{B}(\vec{s} \mid \vec{o})$	$STTT$	SL_HTT	$SLTT$	$SITT$	SIL_HT	$SILT$	$SIWT$	SIW_HT
$o_\emptyset o_\emptyset o_\emptyset o_\emptyset$	$1 - p'_H$	p'_H						
$o_\emptyset o_L o_\emptyset o_\emptyset$			1					
$o_\emptyset o_I o_\emptyset o_\emptyset$				$1 - p_H$	p_H			
$o_\emptyset o_I o_L o_\emptyset$						1		
$o_\emptyset o_I o_W o_\emptyset$							p_W	$1 - p_W$

(empty cells are 0).

Interpretation. Imposing $p'_H = p_H$ as agreed, the belief is parameterized by the two numbers $p_H, p_W \in (0, 1)$. We will see in Exercise 1.7 that p_W does not enter any quantity of interest; $p_H \in (0, 1)$ is the human's suspicion that an unexplained empty log hides a failed CUDA install.

Definition 1.3 (Observation return)**.** The **observation return** of a state trajectory $\vec{s}$ is the expected true return under the human's belief about which trajectory produced the same observations:

Observation return

$$G_{\text{obs}}(\vec{s}) \;=\; \mathbb{E}_{\vec{s}' \sim \mathcal{B}(\cdot \mid \vec{O}(\vec{s}))}\big[G(\vec{s}')\big].$$

The **observation value** of a policy π is $J_{\text{obs}}(\pi) = \mathbb{E}_{\vec{s} \sim P^\pi}[G_{\text{obs}}(\vec{s})]$. Naive RLHF, given Boltzmann-rational human feedback over trajectory pairs, will select the policy that maximizes J_{obs} in the infinite data limit (rather than the true value J); see Lang et al. [1, Proposition 4.1] for the precise statement.

Exercise 1.6. (Conceptual) Give an interpretation of $J_{\text{obs}}(\pi)$ in terms of what the human *believes* is happening. Why is it natural to say that "RLHF rewards policies for for what their behavior looks like, not for what they do"?

Solution to Exercise 1.6

For a single trajectory $\vec{s}$, the human sees $\vec{O}(\vec{s})$ and, not knowing which underlying trajectory produced it, their best guess of the return is the expectation over their posterior:

$$G_{\text{obs}}(\vec{s}) \;=\; \mathbb{E}_{\vec{s}' \sim \mathcal{B}(\cdot \mid \vec{O}(\vec{s}))}[G(\vec{s}')].$$

This is what the human *believes* the return of $\vec{s}$ to be. The policy-level quantity $J_{\text{obs}}(\pi)$ is then the on-policy average of these trajectory-level beliefs:

$$J_{\text{obs}}(\pi) \;=\; \mathbb{E}_{\vec{s} \sim P^{\pi}}[G_{\text{obs}}(\vec{s})].$$

The catch is what happens when two distinct trajectories produce the same observations. Suppose $\vec{s}$ and $\vec{s}'$ have $\vec{O}(\vec{s}) = \vec{O}(\vec{s}') = \vec{o}$, but $\vec{s}$ has low true return and $\vec{s}'$ has high true return. Then by definition, both trajectories yield the *same* observation return:

$$G_{\text{obs}}(\vec{s}) \;=\; G_{\text{obs}}(\vec{s}') \;=\; \mathcal{B}(\vec{s} \mid \vec{o})\, G(\vec{s}) + \mathcal{B}(\vec{s}' \mid \vec{o})\, G(\vec{s}'),$$

which is a convex combination strictly larger than $G(\vec{s})$ whenever $\mathcal{B}(\vec{s}' \mid \vec{o}) > 0$. So if an agent can arrange for the *actual* trajectory to be the bad $\vec{s}$ while still producing observations $\vec{o}$ shared with a good $\vec{s}'$, the human's belief inflates the apparent return: $G_{\text{obs}}(\vec{s}) > G(\vec{s})$, and this inflation flows through to $J_{\text{obs}}(\pi)$ via the on-policy average.

This is why we say RLHF rewards policies for what their behavior *looks like*, not for what it does: the optimization target J_{obs} cannot distinguish between two policies whose on-policy observation distributions match, even if their on-policy true returns differ arbitrarily. The remaining problems make this concrete in the CUDA example, culminating in a regime where the RLHF-optimal policy is strictly worse under the true reward than another available policy.

Exercise 1.7. Compute $G_{\text{obs}}(\vec{s})$ for each of the eight trajectories in Exercise 1.2, expressing your answers in terms of p_H and r. Verify that for the pair $(SIWT, SIW_HT)$, the parameter p_W indeed does not appear, justifying its omission going forward.

Solution to Exercise 1.7

For trajectories $\vec{s}$ with a unique observation, $G_{\text{obs}}(\vec{s}) = G(\vec{s})$. So $G_{\text{obs}}(SLTT) = -5$ and $G_{\text{obs}}(SILT) = -4$.

For the colliding pairs, G_{obs} is the same for both members and equals the expected true return under the belief:

$$G_{\text{obs}}(STTT) = G_{\text{obs}}(SL_HTT) = (1-p_H)\cdot 0 + p_H\cdot(-5-r) = -p_H(5+r),$$
$$G_{\text{obs}}(SITT) = G_{\text{obs}}(SIL_HT) = (1-p_H)\cdot 1 + p_H\cdot(-4-r) = 1-p_H(5+r),$$
$$G_{\text{obs}}(SIWT) = G_{\text{obs}}(SIW_HT) = p_W\cdot 11 + (1-p_W)\cdot 11 = 11.$$

The last line shows p_W drops out, as anticipated.

Exercise 1.8. Compute the true value $J(\pi)$ and the observation value $J_{\text{obs}}(\pi)$ of the four policies

$$[a_T], \quad [a_I a_T], \quad [a_I a_C a_T], \quad [a_I a_H a_T].$$

Express your answers in terms of p, p_H, and r.

Solution to Exercise 1.8

The on-policy distributions are deterministic up to the stochastic CUDA outcome:

- $[a_T]$: $P^\pi(STTT) = 1$.
- $[a_I a_T]$: $P^\pi(SITT) = 1$.
- $[a_I a_C a_T]$: $P^\pi(SIWT) = p$, $P^\pi(SILT) = 1-p$.
- $[a_I a_H a_T]$: $P^\pi(SIW_HT) = p$, $P^\pi(SIL_HT) = 1-p$.

Taking expectations of G and G_{obs} trajectory-by-trajectory and using the previous exercise gives:

π	$J(\pi)$	$J_{\text{obs}}(\pi)$
$[a_T]$	0	$-p_H(5+r)$
$[a_I a_T]$	1	$1-p_H(5+r)$
$[a_I a_C a_T]$	$15p-4$	$15p-4$
$[a_I a_H a_T]$	$(15+r)p-4-r$	$11p+(1-p)\big(1-p_H(5+r)\big)$

The non-trivial cell is the bottom-right: under a_H, a failure produces trajectory SIL_HT which the human cannot distinguish from $SITT$, so its G_{obs} is $1-p_H(5+r)$ rather than $-4-r$. Likewise the success trajectory SIW_HT has $G_{\text{obs}} = 11$, so

$$J_{\text{obs}}([a_I a_H a_T]) \;=\; 11p + (1-p)\big(1-p_H(5+r)\big).$$

Exercise 1.9. Suppose $p > \frac{1}{3}$, so that $J([a_I a_C a_T]) > J([a_I a_T]) > 0$ and the true-optimal policy among the four is $\pi^* = [a_I a_C a_T]$. Show that under the additional condition

$$p_H < \frac{5}{5+r},$$

the RLHF-optimal policy (the J_{obs}-maximizer) is instead $\pi^{\text{RLHF}} = [a_I a_H a_T]$.

Hint: Compare $J_{\text{obs}}([a_I a_H a_T])$ to $J_{\text{obs}}([a_I a_C a_T])$ and solve for the condition on p_H. Then check the other two policies are dominated.

Solution to Exercise 1.9

First, compare the two "contested" policies:

$$\begin{aligned} J_{\text{obs}}([a_I a_H a_T]) - J_{\text{obs}}([a_I a_C a_T]) &= \big[11p + (1-p)(1 - p_H(5+r))\big] - \big[15p - 4\big] \\ &= (1-p)\big(5 - p_H(5+r)\big). \end{aligned}$$

Since $p < 1$, this is positive iff $p_H(5+r) < 5$, i.e. $p_H < 5/(5+r)$. So under the assumed condition, $J_{\text{obs}}([a_I a_H a_T]) > J_{\text{obs}}([a_I a_C a_T])$.
To see that this beats the remaining two policies as well, observe a clean separation: both contested policies have $J_{\text{obs}} > 1$, while both non-contested policies have $J_{\text{obs}} < 1$.

- $J_{\text{obs}}([a_I a_C a_T]) = 15p - 4 > 1$, using $p > 1/3$; and $J_{\text{obs}}([a_I a_H a_T]) > J_{\text{obs}}([a_I a_C a_T]) > 1$ by the comparison above.
- $J_{\text{obs}}([a_I a_T]) = 1 - p_H(5+r) < 1$ since $p_H > 0$, and $J_{\text{obs}}([a_T]) = -p_H(5+r) < 0 < 1$.

So $[a_I a_H a_T]$ dominates all three alternatives in J_{obs}.
Thus when $p_H < 5/(5+r)$, naive RLHF selects $\pi^{\text{RLHF}} = [a_I a_H a_T] \neq [a_I a_C a_T] = \pi^*$.

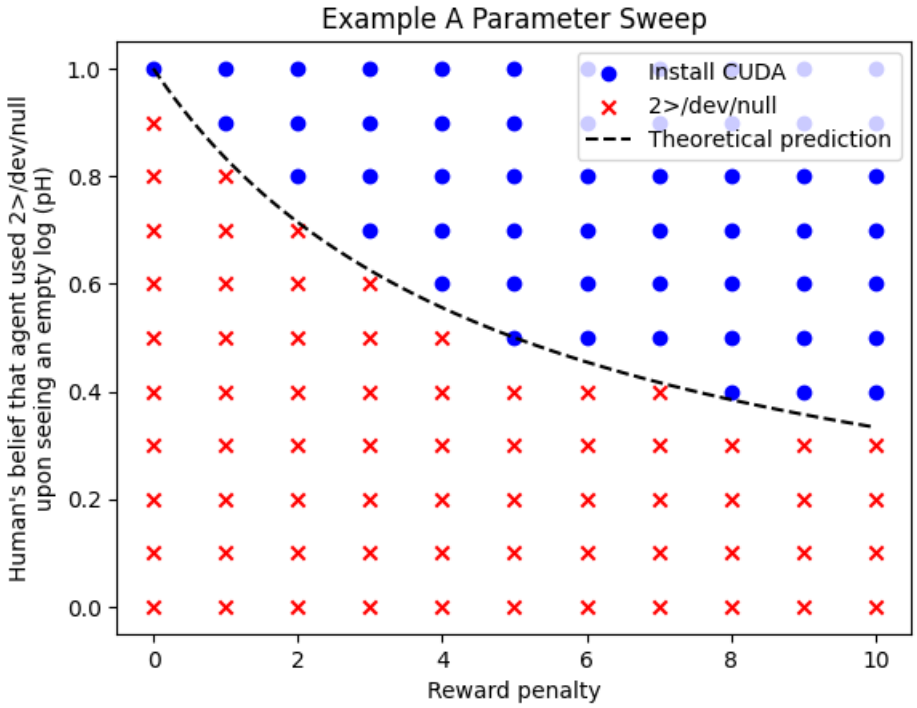

Figure 2: Empirical sweep over (r, p_H) for the CUDA example, with p fixed in the $p > 1/3$ regime [1, Figure 5A]. Each cell reports the action selected at state I by an RLHF-trained policy; "hide" is a_H and "default logging" is a_C.

Exercise 1.10. (Discussion) Figure 2 shows the empirical behavior of an RLHF-trained policy across the (r, p_H) plane (with p fixed in the $p > 1/3$ regime studied above). Discuss:

(a) Where in the plane does the figure show the agent switching from a_H to a_C? How does this empirical boundary compare to the analytical threshold $p_H = 5/(5 + r)$ derived in Exercise 1.9?

(b) For $r = 0$ (no penalty for hidden errors), does any value of $p_H \in (0, 1)$ make a_C the RLHF-optimal action? What does this say about RLHF's ability to discourage `/dev/null` in the absence of an explicit user preference against it?

Solution to Exercise 1.10

(1) The empirical switch from "hide" to "default logging" in Figure 2 traces out exactly the curve $p_H = 5/(5 + r)$ derived in Exercise 1.9: above the curve (high p_H, the human is suspicious enough), a_C is RLHF-optimal; below it (low p_H, the human is too trusting of empty logs), a_H wins.

(2) For $r = 0$, the threshold becomes $p_H < 5/5 = 1$, which holds for every $p_H \in (0, 1)$. So whenever $r = 0$, RLHF picks a_H regardless of how suspicious the human is. Intuitively: if the user has no expressed dislike of hidden errors, RLHF has no signal to penalize `/dev/null`, and the agent will exploit the human's uncertainty to inflate its apparent return whenever installation might fail.

What we've learned. In Exercise 1.6 we saw that RLHF rates policies by how they *look*, not by what they do. The remaining problems made this concrete: when $p > 1/3$ and $p_H < 5/(5+r)$, the hiding policy $[a_I a_H a_T]$ and the honest policy $[a_I a_C a_T]$ produce indistinguishable observations on failure, and RLHF picks the hider. This is *deceptive inflation*: the agent exploits the human's uncertainty to inflate the perceived return [1, Section 4].

There is a dual failure mode — in the same MDP, opposite regime. When p is small (CUDA isn't worth attempting) and p_H is large (the human strongly suspects every empty log of hiding a failure), the honest policy $[a_I a_T]$ produces an empty post-driver log that *looks* like a hidden failure to the suspicious human. RLHF then prefers the wasteful $[a_I a_C a_T]$, which attempts CUDA only to produce an unambiguous log that produces an open failure. The agent pays real reward to prove its honesty: this is *overjustification* [1, Section 4].

The bigger picture. The CUDA example is small, but the phenomenon is general: any alignment-by-feedback method (RLHF, RLAIF, constitutional methods, ...) grades the agent by *whatever the evaluator can tell*, not by what is true. Partial observability is one cause of that gap, but limits on expertise, attention, or time produce the same trap. This is the motivation for proposals like AI safety via debate (which we discussed yesterday): two AIs argue in front of a human judge, each pointing out flaws in the other, in the hope that the judge reaches a correct conclusion they could not reach unaided. Whether debate actually escapes the trap remains open; the point is that any feedback-based method has to confront the gap between "what the human can tell" and "what is true."

Learn more

Here we list further readings, which are largely papers mentioned in Leon's lecture slides.

- Faulty reward functions in the wild: A basic reward specification problem.
- Reward Learning methods and frameworks:
  - Deep reinforcement learning from human preferences — the most basic and popular approach to reward learning

  - Reward-rational (implicit) choice: A unifying formalism for reward learning — a generalization that contains RLHF as a special case
    * Algorithms for Inverse Reinforcement Learning: Another special case
    * Preferences Implicit in the State of the World: Another special case
    * Cooperative Inverse Reinforcement Learning: A *generalization*
    * Benefits of Assistance over Reward Learning: An adapted framework for said generalization

- Underspecification and misspecification
  - Occam's razor is insufficient to infer the preferences of irrational agents: This is a case of an underspecification of the relationship between the human's reward function and the human's policy
  - When Your AIs Deceive You: Challenges of Partial Observability in Reinforcement Learning from Human Feedback: In this misspecification, humans are assumed to fully observe the environment even if they only do so partially.
  - Modeling Human Beliefs about AI Behavior for Scalable Oversight: An approach to correct for the previous misspecification via human models; this can then, however, sometimes lead to an *underspecification* in which the reward function is too uncertain to be safely learned.
  - AI Alignment with Changeable and Influenceable Reward Functions: This paper breaks with the typical assumption of a fixed reward function.
  - Social Choice Should Guide AI Alignment in Dealing with Diverse Human Feedback: This paper breaks with the typical assumption of a *single* reward function.
    * Potential answer: Collective Constitutional AI: Aligning a Language Model with Public Input

- On how reward learning falls within AI alignment:
  - Alignment targets
    * OpenAI Model Spec
    * Claude's new constitution
    * Coherent Extrapolated Volition
  - The reward learning agenda assumes that human values can be expressed via reward functions. However, some papers argue:
    * The Reward Hypothesis is False
  - If human values are captured by a reward function, it is still contentious that we should attempt to *decompose* the alignment problem into first learning said reward function and then optimizing it:

        * Inner and outer alignment decompose one hard problem into two extremely hard problems
        * Reward is not the optimization target
    - Even if we'd solve outer alignment via reward learning, we'd still be left with the inner alignment problem of finding a policy that "cares for" this objective:
        * Risks from Learned Optimization in Advanced Machine Learning Systems
        * Reinforcement Learning textbook: This book is on reinforcement learning, which can be regarded as a very basic conceptualization of the *inner* alignment problem
        * Goal misgeneralization in Deep Reinforcement Learning

- Learn more also in Joar Skalse's sequence on the theoretical foundations of reward learning

- I can also recommend reading the work of Anca Dragan and her many students on reward learning.

References

[1] Leon Lang, Davis Foote, Stuart Russell, Anca Dragan, Erik Jenner, and Scott Emmons. When your AIs deceive you: Challenges of partial observability in reinforcement learning from human feedback. In *Advances in Neural Information Processing Systems*, 2024.