Active Teacher Selection for Reinforcement Learning from Human Feedback

Anonymous submission

Abstract

Reinforcement learning from human feedback (RLHF) enables machine learning systems to learn objectives from human feedback. A core limitation of these systems is their assumption that all feedback comes from a single human teacher, despite querying a range of distinct teachers. We propose the *Hidden Utility Bandit* (HUB) framework to model differences in teacher rationality, expertise, and costliness, formalizing the problem of learning from multiple teachers. We develop a variety of solution algorithms and apply them to two real-world domains: paper recommendation systems and COVID-19 vaccine testing. We find that the *Active Teacher Selection* (ATS) algorithm outperforms baseline algorithms by actively selecting when and which teacher to query. The HUB framework and ATS algorithm demonstrate the importance of leveraging differences between teachers to learn accurate reward models, facilitating future research on active teacher selection for robust reward modeling.

1 Introduction

Specifying objective functions for machine learning systems is challenging, and misspecified objectives can be hacked (Pan, Bhatia, and Steinhardt 2022; Skalse et al. 2022) or incentivise degenerate behavior (Zhuang and Hadfield-Menell 2020; Thomas and Uminsky 2020; Krakovna et al. 2020). Techniques such as *reinforcement learning from human feedback* (RLHF) enable ML systems to instead *learn* appropriate objectives from human feedback (Christiano et al. 2017; Lee, Smith, and Abbeel 2021; Stiennon et al. 2020). These techniques are widely used to finetune large language models (OpenAI 2023; Anthropic 2023; Touvron et al. 2023; Google 2023) and to train reinforcement learning agents (Christiano et al. 2017; Lee, Smith, and Abbeel 2021). However, while RLHF is relied upon to ensure that these systems are safe, helpful, and harmless (Bai et al. 2022), it still faces many limitations and unsolved challenges (Casper et al. 2023).

In particular, RLHF systems typically rely on the assumption that all feedback comes from a single human teacher, despite gathering feedback from a range of teachers with varying levels of rationality and expertise (Daniels-Koch and Freedman 2022). For example, Stiennon et al. (2020), Bai et al. (2022) and Ouyang et al. (2022) assume that all feedback comes from a single teacher, but find that annotators and researchers actually disagree 23% to 37% of the time. Reward learning has been shown to be highly sensitive to incorrect assumptions about the process that generates feedback (Hong, Bhatia, and Dragan 2022; Freedman, Shah, and Dragan 2021; Skalse and Abate 2022; Milli and Dragan 2020), so this single-teacher assumption exposes these systems to dangerous failures (Daniels-Koch and Freedman 2022). Ideally, RLHF systems should consider differences in teacher rationality and expertise to improve their safety and reliability.[1]

We introduce a novel problem called a *Hidden Utility Bandit (HUB)* to model teacher differences in RLHF. A HUB is similar to a multi-armed bandit (MAB), in that at each timestep the agent has a consistent set of alternatives (called "arms") and receives utility based on which it chooses ("pulls"). Unlike a MAB, however, the agent observes the arm's output ("item") *but not the associated utility*. Like in RLHF, it must learn the utility function based on comparison feedback, but unlike in RLHF, the agent can choose amongst multiple teachers. Optimal HUB solutions must therefore actively select *which* teachers to query *when* so as to maximize the expected discounted sum of utilities. Figure 1 shows a simple HUB in which the two arms are vending machines, the two teachers are human taste-testers, and the outputs are fruit.

We present preliminaries in Section 2 (and discuss further related work in Section 6), then formalize the HUB framework (Section 3). We then develop an *Active Teacher Selection* (ATS) method that selects *which* teachers to query *when* to maximize cumulative discounted utility (Section 4). Since there are no existing solutions to the novel HUB problem, we introduce multiple families of baseline methods and evaluate these against ATS on a realistic recommendation task (Section 5.1). ATS outperforms methods with fixed exploration windows, demonstrating the usefulness of selecting *when* to query teachers, and ATS with specific teacher selection outperforms general teacher selection, underscoring the usefulness of selecting *which* teacher to query. As a proof-of-concept, we also demonstrate application of this framework to the real-world problem of evaluating COVID-19 vaccines with expensive and unreliable tests (Section 5.2). The result

[1] Of course, teachers may have varying *values* as well. Balancing competing values is an open problem in social choice theory, and thus beyond the scope of this work, but see Conitzer et al. (2024) for further discussion.

is a HUB framework and an ATS algorithm[2] that demonstrate the importance of leveraging differences between teachers to learn accurate reward models. These will facilitate and benchmark improved methods, ultimately leading to scalable reward learning algorithms that learn accurate, robust and value-aligned models.

2 Preliminaries

Multi-armed bandits (MAB) are stateless sequential decision-making problems (Robbins 1952; Slivkins 2019). At each timestep the agent chooses one of K arms, each with a distribution over utilities. When the agent pulls arm $k \in K$, it receives utility sampled from arm k's distribution $u \sim \mathcal{D}^k$. The agent's goal is to maximize its expected cumulative utility. Our framework is similar, though arm utilities are hidden (as in many real-life applications), and the agent must learn about them from teacher preferences (as in RLHF).

Partially observable Markov decision processes (POMDP) are sequential decision-making problems where the world state is partially hidden from the agent (Littman, Cassandra, and Kaelbling 1995). A POMDP problem is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{O}, \Omega, \gamma \rangle$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, $\mathcal{T}$ and $\mathcal{R}$ are the transition and reward functions, and γ is the discount factor. At time t, the agent begins in state s_t, takes action a_t, transitions to state s_{t+1} determined by $\mathcal{T}(s_t, a_t)$ and receives reward $r_t = R(s_t, a_t, s_{t+1})$. Rather than observing states directly, the agent observes an observation ω_{t+1} from the observation space $\mathcal{O}$ determined by the observation function $\Omega(s_{t+1}, a_t)$. A POMDP solution is a policy that balances inferring the underlying state and acting in the environment to maximise expected cumulative reward.

While calculating this solution is typically intractable, approximate POMDP algorithms can perform well. Partially observable Monte Carlo planning (POMCP) algorithms produce time-efficient online solvers that form a belief tree of fixed depth then use rollouts to estimate the values of the leaf nodes (Silver and Veness 2010). In this work we use partially observable Monte Carlo planning with observation widening (POMCPOW), a POMCP-style algorithm that uses a weighted particle filter to efficiently produce approximate solutions for problems with large state spaces (Sunberg and Kochenderfer 2018).

3 Hidden Utility Bandits

We design the *Hidden Utility Bandit* (HUB) framework to formalize the problem of reward learning from multiple teachers. Formally, a HUB is a partially-observable sequential decision-making problem consisting of a set of items (each with a distinct utility), a set of arms (each with a fixed distribution over items), and a set of teachers (each with a rationality parameter and cost). We assume that the agent can take one action (pulling an arm or querying a teacher) per timestep. Following an existing standard in work on RLHF (Lee, Smith, and Abbeel 2021), a HUB models each teacher as Boltzmann-rational with its noisiness modulated by a *rationality parameter* $\beta \in [0, \infty)$. In particular, the probability that a teacher with rationality parameter β prefers item i to j is below:

$$\Pr(i \succ j; \beta, \mathcal{U}) = \frac{\exp(\beta \mathcal{U}(i))}{\exp(\beta \mathcal{U}(i)) + \exp(\beta \mathcal{U}(j))}, \qquad (1)$$

where $\mathcal{U} : \mathcal{I} \to \mathbb{R}$ gives the true utility of all items in set $\mathcal{I}$.

At each step of the HUB problem, the agent chooses between pulling an arm, observing an item sampled from that arm's distribution and receiving *but not observing* that item's utility, or querying a teacher, receiving feedback modulated by that teacher's rationality parameter but incurring that teacher's query cost. We assume that all teachers give feedback based on a single shared utility function. The agent's objective is to maximize the expected discounted sum of utilities, so it must balance querying costly teachers to learn about the utility function with pulling arms to earn utility.

Definition 3.1. A ***hidden-utility bandit*** **(HUB)** is a tuple $\langle \mathcal{I}, \mathcal{U}, \mathcal{C}, \beta, F, Q, \gamma \rangle$:

- $\mathcal{I}$ is a set of N *items*, each with a hidden utility.
- $\mathcal{U} : \mathcal{I} \to [u_{\min}, u_{\max}]$ is a *utility function* over $\mathcal{I}$, where $\mathbb{U}$ is the utility function space.
- $\mathcal{C} = \{c^1, c^2, \ldots, c^K\}$ is a set of K *arm choices*, each associated with an *arm distribution* $\mathcal{D}^k : \mathcal{I} \to [0,1]$ giving the probability of returning each item in $\mathcal{I}$, where $\mathbb{D} = \mathbb{D}^1 \times \mathbb{D}^2 \times \cdots \times \mathbb{D}^K$ is the joint arm distribution space over all arm choices $\mathcal{C}$.
- $\beta = \{\beta^1, \beta^2, \ldots, \beta^M\}$ is M *teacher rationality parameters*.
- $F = \{f^1, f^2, \ldots, f^M\}$ is M *teacher query costs*.
- $Q : \mathcal{I} \times \mathcal{I} \to [0,1]$ is a *query profile* that gives probabilities of picking queries in $\binom{\mathcal{I}}{2}$.
- γ is a discount factor.

Here, the agent can observe $\mathcal{I}$, $\mathcal{C}$, β, F, Q, and γ but cannot observe the utility function $\mathcal{U}$ or the arm distributions $\mathcal{D}$. At each timestep t, the agent can select an arm choice $c_t \in \mathcal{C}$ or a teacher rationality parameter $\beta_t \in \beta$. If the agent pulls an arm choice $c_t \in \mathcal{C}$, it observes an item i_t sampled from the arm distribution $\mathcal{D}^{c_t}$ and receives but does *not* observe the utility $u_t = \mathcal{U}(i_t)$. Conversely, if the agent queries a teacher with rationality parameter $\beta_t \in \beta$, it receives and observes an item pair (i, j) sampled from the query profile Q, a preference p_t sampled from $\text{Bernoulli}(P)$ given the probability $P = \Pr(i \succ j; \beta_t, \mathcal{U})$ in Equation 1, and the teacher query cost $u_t = f^{\beta_t}$.

Since the agent's objective is to maximize the expected discounted sum of utilities $\mathbb{E}[\Sigma_{t=0}^{\infty} \gamma^t u_t]$, it must balance querying teachers to learn about the utility function with selecting bandit arms to earn utility. Standard RLHF systems alternate between fitting a reward model to teacher feedback and learning a policy using the reward model on a predefined schedule. However, the HUB framework allows the agent to interweave these processes to optimize performance.

3.1 Naive HUB Inference

We propose a naive HUB inference baseline in Algorithm 1. This allows the agent to infer hidden information: the joint arm distribution $\mathcal{D}^{\mathcal{C}} = (\mathcal{D}^1, \mathcal{D}^2, \ldots, \mathcal{D}^K)$ (common to stochastic multi-armed bandit problems) and utility function $\mathcal{U}$ (unique to the HUB). In Algorithm 1, the agent randomly

[2] Our open-source ATS Julia library is available at github.com/[redacted]/ATS.

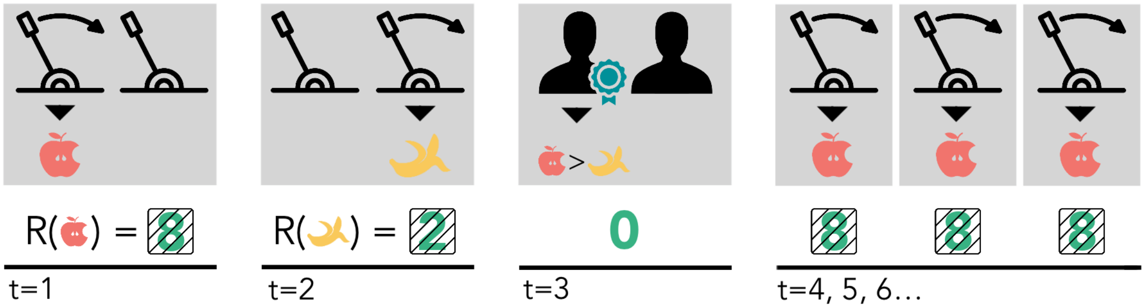


Figure 1: A simple *Hidden Utility Bandit (HUB)* with two arms and two teachers. The agent pulls the first arm, observes an apple, and receives the apple's utility of 8 without observing it. The agent then pulls the second arm, observes a banana, and receives the banana's utility of 2 without observing it. Because these utilities are hidden, the agent foregoes the opportunity for utility on the third timestep to ask the expert teacher which fruit is better. The expert replies that apples are better than bananas, so the agent pulls the first arm to maximize apples for all remaining timesteps.

pulls arms and queries a preselected teacher for a fixed number of timesteps (lines 1-12), approximates the joint arm distribution and teacher preference probabilities with sample means (lines 13-14), then estimates the utility function (lines 15-16). The agent can then simply calculate the the expected utility of each arm and pull the arm with the highest expected utility for the remainder of the episode.

Using these approximate teacher preference probabilities, the *difference* in utility values for items i and j, $\Delta_{ij} = \mathcal{U}(i) - \mathcal{U}(j)$, can be calculated in the following way:

$$\Pr(i \succ j; \beta^m, \mathcal{U}) = \frac{1}{1 + \exp(-\beta^m \Delta_{ij})} \implies \Delta_{ij} \tag{2}$$

$$= -\frac{1}{\beta^m} \ln \left[\frac{1}{\Pr(i \succ j; \beta^m, \mathcal{U})} - 1 \right]. \tag{3}$$

Despite the simplicity of Algorithm 1, it is possible to prove that it converges to the ground truth utility function $\mathcal{U}^*$ and arm distribution set $\mathcal{D}^{\mathcal{C}*}$ in the limit of infinite queries. We prove the following theorem in Appendix A.1:

Theorem 3.2. *If the predicted utility function $\hat{\mathcal{U}}$ and the predicted arm distribution $\hat{\mathcal{D}^{\mathcal{C}}}$ are estimated by executing Algorithm 1 with T samples, then $\hat{\mathcal{U}} \to \mathcal{U}^*$ and $\hat{\mathcal{D}^{\mathcal{C}}} \to \mathcal{D}^{\mathcal{C}*}$ as $T \to \infty$.*

However, exploring randomly for a fixed number of timesteps and querying a fixed teacher may be suboptimal. By maintaining and updating an internal belief over the hidden information, the agent can query teachers only when teacher feedback is necessary to update its belief.

4 Active Teacher Selection

The *Active Teacher Selection (ATS)* algorithm solves the HUB problem efficiently by maintaining a belief over the utility function and arm distributions, and choosing when to query teachers. This allows it to only query when required for decision-relevant belief updates. ATS also actively selects *which* teacher to query. When teachers are "noisy" ($\beta < \infty$), the preference probability $\Pr(i \succ j; \beta, \mathcal{U})$ correlates with the difference in utility between i and j, so it will sometimes be more informative for ATS to select teachers with *lower* β values (Michaud, Gleave, and Russell 2020; Barnett et al. 2023). Importantly, this removes the need to set the problem-specific hyperparameters in Algorithm 1 for exploration (T) and teacher selection (β^m).

4.1 ATS Algorithm

The ATS algorithm has two general steps: the HUB is first converted to a simplified partially observable Markov decision process (POMDP) (Littman, Cassandra, and Kaelbling 1995) and then solved using a Monte Carlo POMDP solver with custom rollout policies.

Constructing the HUB-POMDP The HUB-POMDP state contains the HUB utility function and arm distributions. The HUB-POMDP reward function gives the *expected* utility of each arm according to this state.

Definition 4.1. A **hidden utility bandit POMDP (HUB-POMDP)** is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \Omega, \mathcal{O} \rangle$:

- $\mathcal{S} = \mathbb{U} \times \mathbb{D}$ is the state space: the state $s \in \mathcal{S}$ is a tuple $\langle \mathcal{U}, \mathcal{D}^{\mathcal{C}} \rangle$ that is fixed.
- $\mathcal{A} = \mathcal{C} \cup \beta$ is the action space: the arm choices $\mathcal{C}$ and teachers β.
- $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the stationary transition function: $\mathcal{T}(s, a) = s \; \forall_{s \in \mathcal{S}} \; \forall_{a \in \mathcal{A}}$.
- $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function:

$$\mathcal{R}(s, a) = \begin{cases} \Sigma_{i \in \mathcal{I}} \mathcal{U}(i) \mathcal{D}^a(i) & \text{if } a \in \mathcal{C} \\ -f^a & \text{if } a \in \beta \end{cases}$$

- $\Omega : \mathcal{I} \cup \mathbb{P}$ is the observation space: the items $\mathcal{I}$ and query-preferences $\mathbb{P} = \mathcal{I} \times \mathcal{I} \times \{0, 1\}$.
- $\mathcal{O} : \mathcal{A} \times \Omega \to [0, 1]$ is the observation function:

$$\mathcal{O}(a, \omega) = \begin{cases} \mathcal{D}^a(i) & \text{if } a \in \mathcal{C} \\ Q(i, j) \Pr(i \succ j; \beta^m = a, \mathcal{U}) & \text{if } a \in \beta \end{cases}$$

Teacher selection can be *general* or *specific*. Under specific selection, the agent chooses which teacher to query. The HUB-POMDP's action space contains all M teachers, $\mathcal{A} = \mathcal{C} \cup \beta$, as shown in the HUB-POMDP above. Under general

Algorithm 1: NAIVEHUBINFERENCE($\cdot$)

```
Require: HUB $\langle \mathcal{I}, \mathcal{U}, \mathcal{C}, \beta, F, Q, \gamma \rangle$, $u_{\min}$, $u_{\max}$, $T$ samples, $\beta^m$ of selected teacher
Initialize: frequency[$c$], frequency[$c$][$i$], frequency[$b$][$q$], preferences[$b$][$q$]
for $t = 1, \ldots, T$ do
    if sampleUniformly({TRUE, FALSE}) then
        sample $c \sim \mathcal{C}$                                    ▷ Sample arm uniformly at random
        sample $i \sim \mathcal{D}^c$                                  ▷ Sample item from (unobserved) arm distribution
        frequency[$c$] ← frequency[$c$] + 1
        frequency[$c$][$i$] ← frequency[$c$][$i$] + 1
    else
        sample $b \sim \beta$                                          ▷ Sample teacher uniformly at random
        sample $q = (i, j) \sim Q$                                     ▷ Sample query from query profile
        sample $p \sim$ Bernoulli($\Pr(i \succ j; b, \mathcal{U})$)    ▷ Sample preference given Equation 1
        frequency[$b$][$q$] ← frequency[$b$][$q$] + 1
        preferences[$b$][$q$] ← preferences[$b$][$q$] + $p$
$\hat{D}^c(i) \leftarrow \frac{\text{frequency}[c][i]}{\text{frequency}[c]} \quad \forall c \in \mathcal{C},\, i \in \mathcal{I}$      ▷ Estimate arm distributions
$\hat{P}(b, q) \leftarrow \frac{\text{preferences}[b][q]}{\text{frequency}[b][q]} \quad \forall b \in \beta,\, q \in Q$      ▷ Estimate preference probabilities
$\Delta_{ij} = -\frac{1}{\beta^m} \ln \left[ \frac{1}{\hat{P}(\beta^m,\, q=(i,j))} - 1 \right] \quad \forall i, j \in \mathcal{I}$      ▷ Calculate using Equation 2
$(x, y) \leftarrow \arg\max_{x,y} [\Delta_{xy}]$                       ▷ Find indices of maximum element
$\hat{\mathcal{U}}(y) \leftarrow u_{\min}, \quad \hat{\mathcal{U}}(i) \leftarrow \left[ \frac{u_{\max}}{u_{\max} - u_{\min}} \right] \Delta_{iy} + u_{\min} \quad \forall i \in \mathcal{I} \setminus \{y\}$      ▷ Estimate utilities
```

selection, the agent chooses *when* to query a teacher, but as in RLHF cannot choose *which* teacher to query. The HUB-POMDP's action space is modified to contain a single general teacher selection action, $\mathcal{A} = \mathcal{C} \cup \{\beta^g\}$.

These alternatives offer a tradeoff: general selection reduces the state space size and computational complexity while specific selection provides the agent with additional control over its feedback. Our experimental results (reported in Section 5.1) indicate that specific greatly outperforms general teacher selection, so we will use ATS with specific teacher selection unless otherwise specified.

Solving the POMDP While exact POMDP solutions are typically intractable, approximate POMDP algorithms often perform well. *Partially observable Monte Carlo planning (POMCP)* algorithms produce time-efficient online solvers that form a belief tree of fixed depth and use rollouts to estimate leaf node values (Silver and Veness 2010). *POMCP with observation widening (POMCPOW)* uses a weighted particle filter to efficiently produce approximate solutions for problems with large state spaces (Sunberg and Kochenderfer 2018), so we adapt it to the HUB-POMDP with specialized rollout policies. We describe and compare candidate rollout policies that we designed specifically for the HUB problem in Appendix B. ATS with the custom *best arm* rollout policy performs best, so we use that POMCPOW variant.

4.2 Query Sample Complexity

We evaluate the efficiency of learning hidden utilities $\mathcal{U}$ from teacher queries by computing upper and lower bounds on teacher query sample complexity. Theorem 4.2 (proved in Appendix A.2) upper bounds queries required for a confident estimation as a function of the number of correct classifications required to learn $\mathcal{U}$, which is nontrivial to calculate but can be estimated from our naive baseline experiments (described in Section 4.3). Theorem 4.3 (proved in Appendix A.3) lower bounds queries required to receive correct classifications for each query pair at least once, assuming that the teacher is queried about each pair until a correct answer is received.

Theorem 4.2. *We can bound the maximum number of teacher queries t required to learn the hidden utilities $\mathcal{U}$ with $\overline{\kappa}$ confidence in the worst case as $t \leq \frac{r(1-p)}{p\overline{\kappa}}$, where r is the number of successful classifications from teachers required to learn the hidden utilities, and $p = \min_{\beta, \Delta_{ij}} \frac{1}{1+\exp(-\beta(\Delta_{ij}))}$ is the worst-case teacher accuracy.*

Theorem 4.3. *We can bound the minimum number of teacher queries t required to receive a correct answer for each query pair with $\overline{\kappa}$ confidence in the worst case as $t \geq \frac{\log(\frac{k}{p})N(N-1)}{2\log(1-p)}$, where N is the number of items, and $p = \min_{\beta, \Delta_{ij}} \frac{1}{1+\exp(-\beta(\Delta_{ij}))}$ is the worst-case teacher accuracy.*

4.3 Teacher Noise Inference in ATS

RLHF systems typically assume that the teacher rationality parameters β are known. However, as this is sometimes unrealistic, we show in Theorem 4.4 that β can also be estimated from preference data. Specifically, given $\Pr(i \succ j; \beta_m, \mathcal{U})$, it is possible to etimate $\hat{\beta}_m = \frac{1}{z}\beta_m$, where z is a scaling factor determined by $\mathcal{U}$. z is based on the difference $\Delta_{ij} = \mathcal{U}(i) - \mathcal{U}(j)$, so as long as the same comparison pair (i, j) is used, all teacher rationality estimates will be on the same scale. (They can be calculated directly if Δ_{ij} happens to be known for a specific (i, j).)[3] We prove the theorem below in Appendix A.4.

[3] Note that it is also possible to directly add β to the state space of the HUB-POMDP and then solve it, but this increases the size of the state space and makes the problem less tractable.

Theorem 4.4. *Given two items $i, j \in \mathcal{I}$ where $\mathcal{U}(i) < \mathcal{U}(j)$ and the preference probability $P = \Pr(i \succ j; \beta_m, \mathcal{U})$ from Equation 1 we can estimate $\hat{\beta}_m = \frac{1}{z}\beta_m$ as in Equation 4. If Δ_{ij} is known, we can further calculate $\beta_m = z \cdot \hat{\beta}_m$, where $z = -\Delta_{ij}^{-1}$.*

$$\hat{\beta}_m = \ln\Big(\frac{1}{P} - 1\Big). \tag{4}$$

We demonstrate this procedure in our experiments in Section 5.2. In addition, we evaluate this procedure in simulation by setting $\beta = \{0.01, 1.0\}$, running a random policy for 1000 timesteps, estimating $\{\hat{\beta}_1, \hat{\beta}_2\}$, and scaling the estimate so that the greatest value is equal to 1.0. We observe a mean squared error of only 0.061 across 100 simulations, indicating that this procedure is accurate.

5 Experiments

We apply the HUB framework to two real-world domains: *paper recommendations* and *COVID-19 vaccine testing*. In the recommendation domain, we conduct comprehensive experiments that evaluate the performance of various solution algorithms (Section 5.1), compare rollout simulation policies (Appendix B), and examine the impact of varying teacher query costs (Appendix C). The more complex vaccine domain provides a proof-of-concept, using the HUB framing to address an urgent problem and demonstrating how β values can be estimated from real-world data. We find that the HUB framework captures both problems well, that the ATS algorithm outperforms all baselines in comprehensive testing in the recommendation domain, and that ATS is the best-performing algorithm that also identifies the best vaccine in the vaccine domain proof-of-concept.

Algorithms We fix ATS to use *specific* teacher selection and the *best arm* rollout policy unless otherwise specified. To our knowledge, the HUB problem is novel and has no solutions in prior literature, so we construct multiple families of baseline methods (*naive* and *random*) for comparison. *Naive* algorithms choose randomly amongst pulling arms and querying the selected teacher for T timesteps, use these observations to estimate the arm distributions and utility function (using Algorithm 1), then pull the arm with the highest estimated expected utility at each timestep. Naive algorithms require problem-specific hyperparameters β^m and T, so for these experiments we select the intermediate of 3 teachers ($\beta^m = \beta^2$) and test a range of exploration horizons ($T \in [50, 100, 200]$). *Random* algorithms select actions uniformly at random from a given set. We evaluate a random algorithm that selects actions from the entire action space, as well as one that selects only arms.

5.1 Conference Recommendation Domain

In the recommendation domain, the system recommends AI conferences from which to read relevant papers. There are three paper categories (Application, Benchmark, Theory) with specified relevance scores, and three conferences (ICLR, ICML, AAAI) with different paper category compositions[4]. The recommender cannot directly observe the relevance scores, so it must learn them by asking professors, whose judgements vary from completely random ($\beta^1 = 0$) to highly accurate ($\beta^3 = 50$). In these experiments, query costs are always 0. (See Appendix C for experiments varying query costs.) Each day, the system recommends one conference, a paper is sampled from that conference's distribution, and the system earns a hidden utility score representing that paper's category's relevance. Alternatively, the system queries a professor who provides a preference over a pair of paper categories. Applying the HUB framework, paper categories are the item set $\mathcal{I} = \{A, B, T\}$, relevance scores are the hidden utility function $\mathcal{U}$, conferences are arm choices $\mathcal{C} = \{c^1 = \text{ICLR},\ c^2 = \text{ICML},\ c^3 = \text{AAAI}\}$, and professors are teachers with rationality $\beta = \{\beta^1 = 0, \beta^2 = 0.01, \beta^3 = 50\}$.

Figure 2 shows an example paper recommendation problem in which it will sometimes be more informative to query the noisy Professor 2 over the more rational Professor 3. This is because the frequency with which a noisy teacher prefers a lower-reward item over a higher-reward one gives information about the difference between the rewards, and in this example the recommender must learn *how much* more relevant Application papers are than Benchmark papers. Without this information, the system cannot distinguish between cases where $\mathcal{U}(A) = 8$ (indicating that the expected relevance of ICLR is greater than ICML) and where $\mathcal{U}(A) = 6$ (indicating the reverse).

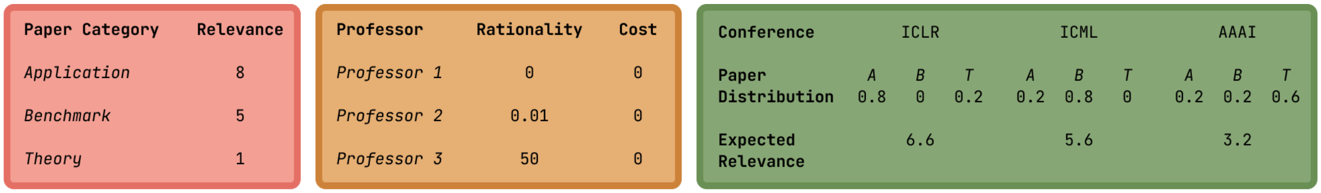
Experiments We evaluate all algorithms for 25 runs of 1000 steps on 20 paper recommendation tasks. Each task is a HUB with $\mathcal{I}$, $\mathcal{C}$, and β as described above and a unique tuple $\langle \mathcal{U}, \mathcal{D}^{\mathcal{C}} \rangle$. $\mathbb{U}$ and $\mathbb{D}$ are discretized, and each task's $\langle \mathcal{U}, \mathcal{D}^{\mathcal{C}} \rangle$ is chosen such that c^1 has the highest expected relevance ($\mathbb{E}[\mathcal{U}(i \sim c^1)] > \mathbb{E}[\mathcal{U}(i \sim c^2)] \geq \mathbb{E}[\mathcal{U}(i \sim c^3)]$) and all paper distributions are different and non-deterministic ($\mathcal{D}^j \neq \mathcal{D}^k\ \forall_{j,k \in \mathcal{C}}$ and $\mathcal{D}^c(i) \neq 1.0\ \forall_{i \in \mathcal{I}, c \in \mathcal{C}}$).

Results While all non-random algorithms successfully identify the most relevant conference in expectation (Figure 3b), ATS with specific teacher selection best balances querying teachers with recommending papers, achieving the highest average discounted cumulative reward (Figure 3a), and most accurately learning relevance scores (Figure 4).

Figure 3b shows how often each algorithm learns to pull the best HUB arm and therefore recommend the most relevant conference over the course of training. All HUB solution methods (ATS, Naive[50], Naive[100], Naive[200]) successfully identify the most relevant conference, recommending it about three times as often as they would if they were behaving randomly ("Random" baseline, light green line) and about twice as often as if they were blindly recommending conferences ("Random Arms" baseline, dark green line). This indicates that the HUB formalism can be used to accurately represent the paper recommendation problem.

While all solution methods identify the best arm, ATS does so most efficiently, querying teachers sparingly even at the start of the task (Figure 3c) and best optimizing the HUB

[4] Example relevance scores and paper category compositions were selected arbitrarily.

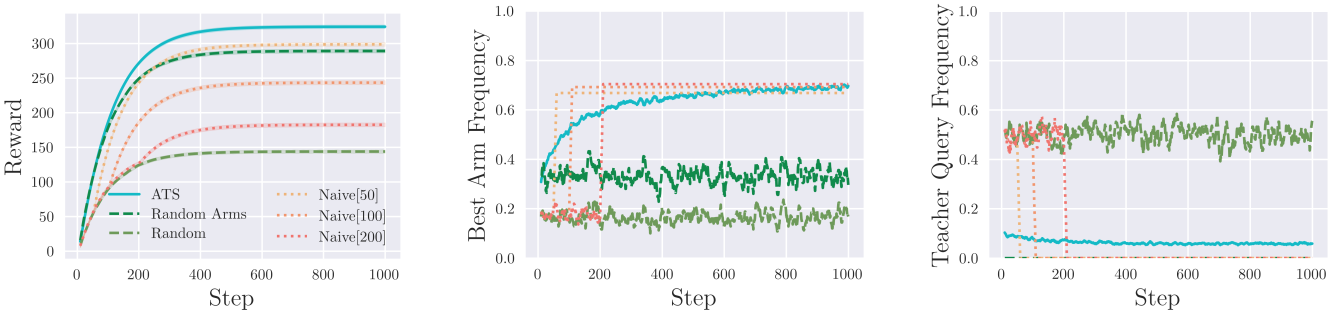


Paper Category	Relevance
Application	8
Benchmark	5
Theory	1

Professor	Rationality	Cost
Professor 1	0	0
Professor 2	0.01	0
Professor 3	50	0

Conference	ICLR			ICML			AAAI		
Paper Distribution	*A* 0.8	*B* 0	*T* 0.2	*A* 0.2	*B* 0.8	*T* 0	*A* 0.2	*B* 0.2	*T* 0.6
Expected Relevance		6.6			5.6			3.2	

Figure 2: Paper recommendation as a HUB problem. Paper categories (Application, Benchmark, Theory) are items ($\mathcal{I}$), professors are teachers with rationality (β) and cost (F) parameters, conferences are arms with distributions ($\mathcal{D}$), and relevance scores are utilities ($\mathcal{U}$). The goal is to recommend the most relevant conferences to read papers from.



(a) Discounted cumulative reward

(b) Frequency of pulling best arm

(c) Frequency of querying teacher

Figure 3: Comparison of ATS, naive and random algorithms. ATS best maximizes discounted reward (a) and identifies the highest-reward arm more often than most baselines and comparably with Naive[100] and Naive[200], which explore more and earn less reward (b). ATS initially queries teachers less often than naive baselines, but continues querying teachers throughout the episode (c). All data is averaged across 25 runs on 20 HUB problems and smoothed over 10 steps.

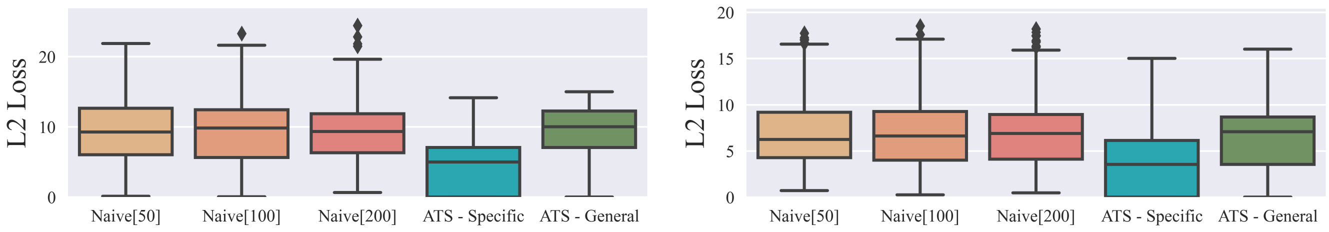
objective of expected discounted cumulative reward (Figure 3a). Moreover, ATS forms the most accurate estimates of the utility function and expected conference relevance scores (Figure 4) after 1000 timesteps, while continuing to explore and potentially improve this estimate by occasionally querying teachers and recommending other conferences (Figure 5a). In contrast, Naive algorithms stop learning after their hand-specified exploration horizon (Figure 5b), and Random algorithms never learn at all (Figure 5c). This demonstrates the benefits of actively selecting when to query teachers, as in ATS, rather than following a predefined RLHF schedule.

Figure 7 compares ATS with specific and general teacher selection. Standard RLHF systems do not allow the agent to select which teacher to query and are most akin to general selection. However, we show that the additional control afforded by specific selection allows ATS to make more informative queries. Figure 7a shows that ATS with specific teacher selection earns higher expected reward than ATS with general teacher selection, and Figure 7b shows that ATS with general teacher selection queries all arms roughly equally, failing to identify the one with highest expected reward.

5.2 COVID-19 Vaccine Testing Domain

Bandit-type problems are commonly used to model medical treatment investigation, so as a proof-of-concept we apply the HUB framework to a real-world medical problem: evaluating vaccines for the 2019 Novel Coronavirus (COVID-19). This task is complicated by the difficulty of evaluating whether a patient is infected: many infections are asymptomatic, and other illnesses cause similar symptoms. There are many ways to test whether patients have COVID-19, including symptom surveys, antigen tests, and RT-PCR tests, but these vary widely in accuracy and cost.

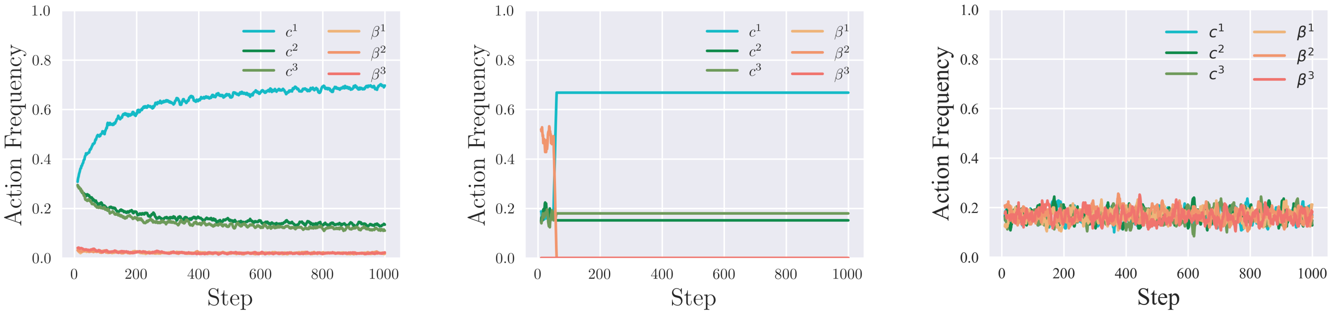
The HUB framework directly models these challenges. Let the item set be easily observable patient symptoms, $\mathcal{I} = \{\text{None}, \text{Cough}, \text{Fever}\}$. The "arms" are vaccine candidates, $\mathcal{C} = \{c^1 = \text{VaccineA}, c^2 = \text{VaccineB}, c^3 = \text{NoVaccine}\}$, and the "teachers" are COVID-19 test types, $\{\text{Survey}, \text{Antigen}, \text{RT-PCR}\}$. Surveys are the least accurate but least expensive, while RT-PCR tests are the most accurate and most expensive. We estimate the US dollar cost of surveys at \$1.20 (accounting for 10 minutes of time at the US federal minimum wage of \$7.25), antigen tests at \$42, and RT-PCR tests at \$62 (median prices reported by (Lo et al. 2023)), then scale these costs by 0.05. We estimate β by gathering real-world data on the sensitivity of COVID-19 symptom surveys (Rufino et al. 2023), antigen tests (Harmon et al. 2021), and RT-PCR tests (Binny et al. 2023), interpret this sensitivity as the probability P of the test "preferring" a patient with no COVID-19 ($\mathcal{U} = u_{max}$) to a patient with definite COVID-19 ($\mathcal{U} = u_{min}$), let $\Delta_{ij} = u_{min} - u_{max}$, and calculate β_m using Equation 4. We construct arm distributions where patients display the most frequent and severe symptoms with no vaccination, and the least symptoms with Vaccine A, and a utility function where symptoms that have a greater chance of indicating COVID-19 infection have lower



(a) $L2$ loss of $\mathcal{U}$ estimate

(b) $L2$ loss of arm reward estimate

Figure 4: Accuracy of reward learning using ATS (with specific and general teacher selection) and naive algorithms (with exploration parameters of 50, 100, and 200). ATS with specific teacher selection learns both the underlying utility function (a) and the expected rewards of each arm (b) much more accurately than ATS with general teacher selection and naive algorithms. The middle line is the median, boxes are the IQR, whiskers are 1.5 times the IQR, and diamonds are outliers.



(a) ATS

(b) Naive[50]

(c) Random

Figure 5: Mean action frequencies for various algorithms. c actions are arm pulls and β actions are teacher queries. Data is averaged across 25 runs of 20 HUB problems and smoothed over 10 steps.

scores. These values are reported in Figure 6.

Experiments We evaluate all algorithms for 25 runs of 1000 steps on this COVID-19 task. $\mathbb{U}$ and $\mathbb{D}$ are more finely discretized than in the recommendation HUB to permit more realistic values, so the resulting HUB-POMDP has 5 times more states and is more challenging to solve. We fix the parameters to the values reported in Figure 6, which are derived from real-world data and estimates.

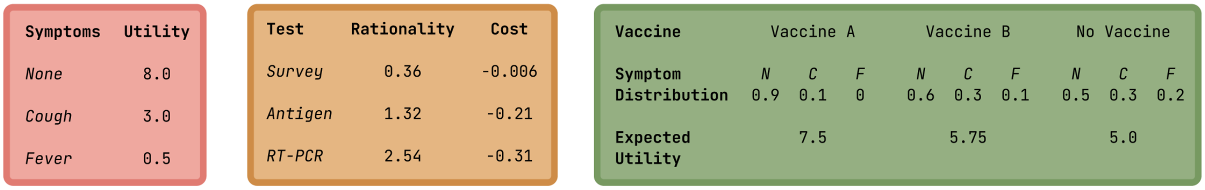
Results Figure 8 summarises the results. ATS is the only algorithm that both earns high reward during training (Figure 8a) and successfully identifies the most effective vaccine (c^1 in Figure 8b). The Random Arms baseline earns slightly higher reward during training (because it vaccinates at every timestep), but fails to learn which vaccine is most effective, which is a core goal of the vaccine trial. The Naive baselines identify the best vaccine, but conduct excessive costly tests.

6 Related Work

Inverse Reinforcement Learning *Inverse reinforcement learning* (IRL) is a reward learning technique in which the agent infers a reward function given behavioral samples from an optimal policy (Ng and Russell 2000; Abbeel and Ng 2004) or a noisy teacher (Ziebart 2010). It is similar to RLHF in that reward information comes from a teacher rather than the environment, but distinct in that it requires teachers to perform the task well themselves (Milli and Dragan 2020). In contrast, RLHF and the HUB framework excel in domains where the teacher can distinguish good performance, but does not know how to produce it themselves, such as those presented in Section 5.

Cooperative Inverse Reinforcement Learning *Cooperative inverse reinforcement learning* (CIRL) extends the IRL framework to allow collaboration between the agent and the teacher (Hadfield-Menell et al. 2016; Malik et al. 2018). HUB problems can be viewed as a specific class of CIRL games in which there are multiple teachers, but they can only act (by providing feedback) when the agent requests it (by querying them). However, CIRL problems are DEC-POMDPS, which are NEXP-complete and thus functionally intractable (Bernstein et al. 2002). By fixing the teacher policy and arm distributions, the HUB framework reduces the problem to a much more tractable POMDP with a stationary transition function. Optimal agent solutions to the CIRL game balance inference and control to produce qualitatively valuable behaviors, such as only asking the human questions when necessary (Shah et al. 2020). The algorithm that best solves the HUB problem, ATS, demonstrates similarly conservative querying behavior.

Crowdsourcing Prior work has investigated the related problem of combining feedback from multiple noisy anno-

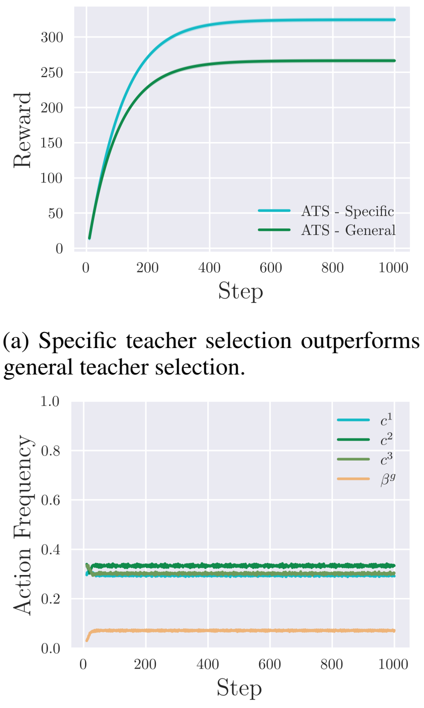


Symptoms	Utility
None	8.0
Cough	3.0
Fever	0.5

Test	Rationality	Cost
Survey	0.36	-0.006
Antigen	1.32	-0.21
RT-PCR	2.54	-0.31

Vaccine	Vaccine A			Vaccine B			No Vaccine		
Symptom Distribution	*N*	*C*	*F*	*N*	*C*	*F*	*N*	*C*	*F*
	0.9	0.1	0	0.6	0.3	0.1	0.5	0.3	0.2
Expected Utility	7.5			5.75			5.0		

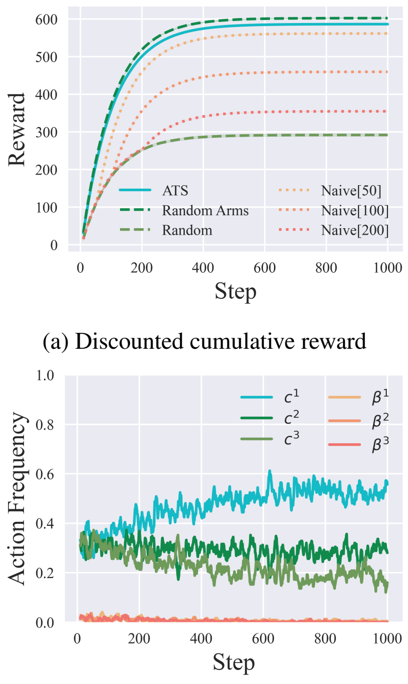
Figure 6: COVID-19 vaccine testing as a HUB problem. Symptoms (None, Cough, Fever) are items ($\mathcal{I}$), tests are teachers with rationality (β) and cost (F) parameters, and vaccines are arms ($\mathcal{C}$) with the specified distributions over patient symptoms ($\mathcal{D}$).



(a) Specific teacher selection outperforms general teacher selection.

(b) ATS with general teacher selection doesn't identify the best arm.

Figure 7: Performance of ATS with specific and general teacher selection. All data is averaged across 25 runs on 20 HUB problems, smoothed over 10 steps, and discounted with $\gamma = 0.99$.



(a) Discounted cumulative reward

(b) ATS action frequencies

Figure 8: Performance of all algorithms and ATS action frequencies on the COVID-19 vaccine testing problem. Random Arms and ATS both earn high reward from frequently vaccinating participants (a), though only ATS additionally identifies the most effective vaccine (b).

tators (Dawid and Skene 1979), often to label training data for supervised learning. Raykar et al. (2010) present an approach that learns teacher expertise and uses teacher feedback to fit a classifier simultaneously, while Rodrigues, Pereira, and Ribeiro (2014) generalise gaussian process classification to model noisy annotators and combine their feedback into reliable labels for supervised learning. This body of work underscores the difficulty and importance of combining feedback from varying and noisy teachers in machine learning.

7 Conclusion

We formalized the teacher selection problem in reward learning and proposed a solution method that expresses this problem as a POMDP. Our empirical results underscore the applicability of this framework to real-world problems, as well as the importance of modeling human teachers as distinct entities and actively choosing *when* and *which* to query.

Limitations and Future Work The purpose of this paper is to investigate the novel problem of selecting teachers in RLHF, so we empirically evaluate tasks where *learning* the utility function is more challenging than *optimizing* it. However, many real-world tasks, such as language model finetuning, also involve challenging optimizations across enormous state spaces. Future work should combine our methods and insights with existing work on state abstractions and hierarchical options to scale the HUB formalism and ATS method to more complex domains.

References

Abbeel, P.; and Ng, A. Y. 2004. Apprenticeship learning via inverse reinforcement learning. In *21st International Conference on Machine Learning*, 1.

Anthropic. 2023. Introducing Claude.

Bai, Y.; Jones, A.; Ndousse, K.; Askell, A.; Chen, A.; DasSarma, N.; Drain, D.; Fort, S.; Ganguli, D.; Henighan, T.; et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Barnett, P.; Freedman, R.; Svegliato, J.; and Russell, S. 2023. Active Reward Learning from Multiple Teachers. In *AAAI Workshop on Artificial Intelligence Safety*.

Bernstein, D. S.; Givan, R.; Immerman, N.; and Zilberstein, S. 2002. The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research*, 27(4): 819–840.

Binny, R. N.; Priest, P.; French, N. P.; Parry, M.; Lustig, A.; Hendy, S. C.; Maclaren, O. J.; Ridings, K. M.; Steyn, N.; Vattiato, G.; et al. 2023. Sensitivity of reverse transcription polymerase chain reaction tests for severe acute respiratory syndrome coronavirus 2 through time. *The Journal of Infectious Diseases*, 227(1): 9–17.

Casper, S.; Davies, X.; Shi, C.; Gilbert, T. K.; Scheurer, J.; Rando, J.; Freedman, R.; Korbak, T.; Lindner, D.; Freire, P.; Wang, T.; Marks, S.; Segerie, C.-R.; Carroll, M.; Peng, A.; Christoffersen, P.; Damani, M.; Slocum, S.; Anwar, U.; Siththaranjan, A.; Nadeau, M.; Michaud, E. J.; Pfau, J.; Krasheninnikov, D.; Chen, X.; Langosco, L.; Hase, P.; Biyik, E.; Dragan, A.; Krueger, D.; Sadigh, D.; and Hadfield-Menell, D. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*.

Christiano, P.; Leike, J.; Brown, T. B.; Martic, M.; Legg, S.; and Amodei, D. 2017. Deep reinforcement learning from human preferences. *Advances in Neural Information Processing Systems*.

Conitzer, V.; Freedman, R.; Heitzig, J.; Holliday, W. H.; Jacobs, B. M.; Lambert, N.; Mossé, M.; Pacuit, E.; Russell, S.; Schoelkopf, H.; et al. 2024. Social Choice for AI Alignment: Dealing with Diverse Human Feedback. *arXiv preprint arXiv:2404.10271*.

Daniels-Koch, O.; and Freedman, R. 2022. The Expertise Problem: Learning from Specialized Feedback. In *NeurIPS Workshop on ML Safety*.

Dawid, A. P.; and Skene, A. M. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1): 20–28.

Freedman, R.; Shah, R.; and Dragan, A. 2021. Choice set misspecification in reward inference. *arXiv preprint arXiv:2101.07691*.

Google. 2023. Bard.

Hadfield-Menell, D.; Russell, S. J.; Abbeel, P.; and Dragan, A. 2016. Cooperative inverse reinforcement learning. *Advances in Neural Information Processing Systems*, 29.

Harmon, A.; Chang, C.; Salcedo, N.; Sena, B.; Herrera, B. B.; Bosch, I.; and Holberger, L. E. 2021. Validation of an athome direct antigen rapid test for COVID-19. *JAMA Network Open*, 4(8): e2126931–e2126931.

Hong, J.; Bhatia, K.; and Dragan, A. 2022. On the Sensitivity of Reward Inference to Misspecified Human Models. *arXiv preprint arXiv:2212.04717*.

Krakovna, V.; Uesato, J.; Mikulik, V.; Rahtz, M.; Everitt, T.; Kumar, R.; Kenton, Z.; Leike, J.; and Legg, S. 2020. Specification gaming: the flip side of AI ingenuity. *DeepMind Blog*.

Lee, K.; Smith, L.; and Abbeel, P. 2021. PEBBLE: Feedback-efficient Interactive Reinforcement Learning via Relabeling Experience and Unsupervised Pre-training. *arXiv preprint arXiv:2106.05091*.

Littman, M. L.; Cassandra, A. R.; and Kaelbling, L. P. 1995. Learning policies for partially observable environments: Scaling up. In *Journal of Machine Learning Research*, 362–370. Elsevier.

Lo, J.; Amin, K.; Telesford, I.; Dawson, L.; and Kates, J. 2023. Prices for COVID-19 testing.

Malik, D.; Palaniappan, M.; Fisac, J.; Hadfield-Menell, D.; Russell, S.; and Dragan, A. 2018. An efficient, generalized Bellman update for cooperative inverse reinforcement learning. In *International Conference on Machine Learning*, 3394–3402. PMLR.

Michaud, E. J.; Gleave, A.; and Russell, S. 2020. Understanding learned reward functions. *arXiv preprint arXiv:2012.05862*.

Milli, S.; and Dragan, A. 2020. Literal or pedagogic human? Analyzing human model misspecification in objective learning. *Uncertainty in Artificial Intelligence*.

Ng, A. Y.; and Russell, S. 2000. Algorithms for inverse reinforcement learning. In *International Conference on Machine Learning*, volume 1, 2.

OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744.

Pan, A.; Bhatia, K.; and Steinhardt, J. 2022. The effects of reward misspecification: Mapping and mitigating misaligned models. *arXiv preprint arXiv:2201.03544*.

Raykar, V. C.; Yu, S.; Zhao, L. H.; Valadez, G. H.; Florin, C.; Bogoni, L.; and Moy, L. 2010. Learning from crowds. *Journal of machine learning research*, 11(4).

Robbins, H. 1952. Some aspects of the sequential design of experiments. *American Mathematical Society*, 58(5): 527–535.

Rodrigues, F.; Pereira, F.; and Ribeiro, B. 2014. Gaussian process classification and active learning with multiple annotators. In *International conference on machine learning*, 433–441. PMLR.

Rufino, J.; Baquero, C.; Frey, D.; Glorioso, C. A.; Ortega, A.; Reščič, N.; Roberts, J. C.; Lillo, R. E.; Menezes, R.; Champati, J. P.; et al. 2023. Using survey data to estimate the impact of the omicron variant on vaccine efficacy against COVID-19 infection. *Scientific Reports*, 13(1): 900.

Shah, R.; Freire, P.; Alex, N.; Freedman, R.; Krasheninnikov, D.; Chan, L.; Dennis, M. D.; Abbeel, P.; Dragan, A.; and Russell, S. 2020. Benefits of assistance over reward learning. *NeurIPS Workshop on Cooperative AI*.

Silver, D.; and Veness, J. 2010. Monte-Carlo Planning in Large POMDPs. *Advances in Neural Information Processing Systems*, 23.

Skalse, J.; and Abate, A. 2022. Misspecification in Inverse Reinforcement Learning. *arXiv preprint arXiv:2212.03201*.

Skalse, J.; Howe, N. H.; Krasheninnikov, D.; and Krueger, D. 2022. Defining and characterizing reward hacking. *arXiv preprint arXiv:2209.13085*.

Slivkins, A. 2019. Introduction to multi-armed bandits. *Foundations and Trends in Machine Learning*, 12(1-2): 1–286.

Stiennon, N.; Ouyang, L.; Wu, J.; Ziegler, D.; Lowe, R.; Voss, C.; Radford, A.; Amodei, D.; and Christiano, P. F. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33: 3008–3021.

Sunberg, Z.; and Kochenderfer, M. 2018. Online algorithms for POMDPs with continuous state, action, and observation spaces. In *International Conference on Automated Planning and Scheduling*, volume 28, 259–263.

Thomas, R.; and Uminsky, D. 2020. The problem with metrics is a fundamental problem for AI. *arXiv preprint arXiv:2002.08512*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; Bikel, D.; Blecher, L.; Ferrer, C. C.; Chen, M.; Cucurull, G.; Esiobu, D.; Fernandes, J.; Fu, J.; Fu, W.; Fuller, B.; Gao, C.; Goswami, V.; Goyal, N.; Hartshorn, A.; Hosseini, S.; Hou, R.; Inan, H.; Kardas, M.; Kerkez, V.; Khabsa, M.; Kloumann, I.; Korenev, A.; Koura, P. S.; Lachaux, M.-A.; Lavril, T.; Lee, J.; Liskovich, D.; Lu, Y.; Mao, Y.; Martinet, X.; Mihaylov, T.; Mishra, P.; Molybog, I.; Nie, Y.; Poulton, A.; Reizenstein, J.; Rungta, R.; Saladi, K.; Schelten, A.; Silva, R.; Smith, E. M.; Subramanian, R.; Tan, X. E.; Tang, B.; Taylor, R.; Williams, A.; Kuan, J. X.; Xu, P.; Yan, Z.; Zarov, I.; Zhang, Y.; Fan, A.; Kambadur, M.; Narang, S.; Rodriguez, A.; Stojnic, R.; Edunov, S.; and Scialom, T. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288.

Zhuang, S.; and Hadfield-Menell, D. 2020. Consequences of Misaligned AI. In *Advances in Neural Information Processing Systems*, 15763–15773.

Ziebart, B. D. 2010. *Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy*. Ph.D. thesis, Carnegie Mellon University.

A Proofs

A.1 State Estimation (Theorem 3.2)

Theorem 3.2. If the predicted utility function $\hat{\mathcal{U}}$ and the predicted arm distribution $\hat{\mathcal{D}^{\mathcal{C}}}$ are estimated by executing Algorithm 1 with T samples, then $\hat{\mathcal{U}} \to \mathcal{U}^*$ and $\hat{\mathcal{D}^{\mathcal{C}}} \to \mathcal{D}^{\mathcal{C}*}$ as $T \to \infty$.

Proof (Sketch).. Since the number of arms is finite and they are pulled uniformly as $T \to \infty$, the number of times that a given arm c^k is pulled approaches infinity. Since each pull samples an item from the true distribution $\mathcal{D}^{k*}$ i.i.d., the empirical distribution $\hat{\mathcal{D}}^k$ will approach $\mathcal{D}^{k*}$ in the limit of infinite pulls. This argument applies for all arms $c^k \in \mathcal{C}$, so $\hat{\mathcal{D}^{\mathcal{C}}} \to \mathcal{D}^{\mathcal{C}*}$ as $T \to \infty$. Similarly, in the limit of infinite queries, $\hat{P}(\beta,(i,j))$ will approach $P^*(\beta,(i,j)) = \Pr(i \succ j; \beta, \mathcal{U}^*)$, the true probability that teacher b prefers item i over item j, as determined by Equation 1. Given β, (i,j) and $\hat{P}(\beta,(i,j))$ from the first T timesteps, we can calculate $\Delta_{ij} = \hat{\mathcal{U}}(i) - \hat{\mathcal{U}}(j)$ using Equation 2. Given $\Delta = [\Delta_{01}, \Delta_{02}, \ldots, \Delta_{NN}]$, u_{max} and u_{min}, we can calculate $\hat{\mathcal{U}}$ as described in Algorithm 1. $\hat{\mathcal{U}} \to \mathcal{U}^*$ as $\hat{P} \to P^*$, which occurs as $T \to \infty$. □

A.2 Query Complexity Upper Bound (Theorem 4.2)

Theorem 4.2. We can bound the maximum number of teacher queries t required to learn the hidden utilities $\mathcal{U}$ with $\overline{\kappa}$ confidence in the worst case as $t \leq \frac{r(1-p)}{p\overline{\kappa}}$, where r is the number of successful classifications from teachers required to learn the hidden utilities, and $p = \min_{\beta,\Delta_{ij}} \frac{1}{1+\exp(-\beta(\Delta_{ij}))}$ is the worst-case teacher accuracy.

Proof. Let $X \sim \text{NegBinom}(r,p)$.

Let $\overline{\kappa}$ represent the upper bound desired confidence.

We bound the maximum number of queries t required to observe r correct classifications with confidence $\overline{\kappa}$ as:

$$\Pr(X = t; r, p) \geq \overline{\kappa}$$

$$\frac{E[X]}{t} \geq \Pr(X = t; r, p) \geq \overline{\kappa} \qquad \text{applying the Markov Inequality}$$

$$\left(\frac{r(1-p)}{p}\right) \cdot \left(\frac{1}{t}\right) \geq \overline{\kappa} \qquad \text{derived from the definition of the Negative Binomial Probability Mass Function}$$

$$\frac{r(1-p)}{p\overline{\kappa}} \geq t$$

The worst case teacher accuracy is

$$\begin{aligned} p &= \min_{\beta,\Delta_{ij}} \frac{\exp(\beta\mathcal{U}(i))}{\exp(\beta\mathcal{U}(i)) + \exp(\beta\mathcal{U}(j))} && \text{from Equation 1} \\ &= \min_{\beta,\Delta_{ij}} \frac{1}{1+\exp(-\beta\Delta_{ij})}. \end{aligned}$$

□

A.3 Query Complexity Lower Bound (Theorem 4.3)

Theorem 4.3. We can bound the minimum number of teacher queries t required to receive a correct answer for each query pair with $\overline{\kappa}$ confidence in the worst case as $t \geq \frac{\log(\frac{k}{p})N(N-1)}{2\log(1-p)}$, where N is the number of items, and $p = \min_{\beta,\Delta_{ij}} \frac{1}{1+\exp(-\beta(\Delta_{ij}))}$ is the worst-case teacher accuracy.

Proof. We bound the minimum number of queries t required to observe the correct answer $i \succ j$ with confidence κ as:

$$\begin{aligned} &\Pr(X \geq t; p) \geq \kappa \\ &(1-p)^t p \geq \kappa \\ &t \geq \frac{\log(\frac{\kappa}{p})}{\log(1-p)} \end{aligned}$$

Since there are N items, there are $\frac{N(N-1)}{2}$ unique pairs. The minimum number of queries t required to observe the correct answer for each pair with confidence κ is therefore

$$t \geq \frac{\log(\frac{k}{p})N(N-1)}{2\log(1-p)}.$$

The worst case teacher accuracy is

$$\begin{aligned} p &= \min_{\beta,\Delta_{ij}} \frac{\exp(\beta\mathcal{U}(i))}{\exp(\beta\mathcal{U}(i)) + \exp(\beta\mathcal{U}(j))} && \text{from Equation 1} \\ &= \min_{\beta,\Delta_{ij}} \frac{1}{1+\exp(-\beta\Delta_{ij})}. \end{aligned}$$

□

A.4 β Estimation (Theorem 4.4)

Theorem 4.4. Given two items $i, j \in \mathcal{I}$ where $\mathcal{U}(i) < \mathcal{U}(j)$ and the preference probability $P = \Pr(i \succ j; \beta_m, \mathcal{U})$ from Equation 1 we can estimate $\hat{\beta}_m = \frac{1}{z}\beta_m$ as in Equation 5. If Δ_{ij} is known, we can further calculate $\beta_m = z \cdot \hat{\beta}_m$, where $z = -\Delta_{ij}^{-1}$.

$$\hat{\beta}_m = \ln\Big(\frac{1}{P} - 1\Big). \tag{5}$$

Proof (Sketch).. First, we define an affine mapping function $f_{a,b}(x) = ax + b$ such that $f_{a,b}(\mathcal{U}(i)) = 0$ and $f_{a,b}(\mathcal{U}(j)) = 1$. Lemma A.1 shows that this is always possible when $\mathcal{U}(i) \neq \mathcal{U}(j)$ and furthermore that $a = \frac{-1}{i-j}$. Let z, y be the parameters that make this mapping for these particular values of $\mathcal{U}(i)$ and $\mathcal{U}(j)$. Note that $z = \frac{-1}{i-j} = -\Delta_{ij}^{-1}$.

Next, suppose we have that $\beta'_m = \frac{1}{a}\beta_m$, it follows that:

$$\begin{aligned} P &= \Pr(i^0 \succ i^1; \beta_m, \mathcal{U}) \\ &= \frac{\exp(\beta_m\mathcal{U}(i))}{\exp(\beta_m\mathcal{U}(i)) + \exp(\beta_m\mathcal{U}(j))} && \text{(by Equation 1)} \\ &= \frac{\exp(\frac{\beta_m}{a} \cdot a\mathcal{U}(i) + \frac{\beta_m}{a} b)}{\exp(\frac{\beta_m}{a} \cdot a\mathcal{U}(i) + \frac{\beta_m}{a} b) + \exp(\frac{\beta_m}{a} \cdot a\mathcal{U}(j) + \frac{\beta_m}{a} b)} \\ &= \frac{\exp(\beta'_m \cdot (a\mathcal{U}(i) + b))}{\exp(\beta'_m \cdot (a\mathcal{U}(i) + b)) + \exp(\beta'_m \cdot (a\mathcal{U}(j) + b))} && \text{(by definition of } \beta'_m) \\ &= \frac{\exp(\beta'_m \cdot f_{a,b}(\mathcal{U}(i)))}{\exp(\beta'_m \cdot f_{a,b}(\mathcal{U}(i))) + \exp(\beta'_m \cdot f_{a,b}(\mathcal{U}(j)))} && \text{(by definition of } f_{a,b}) \\ &= \frac{\exp(0)}{\exp(0) + \exp(\beta'_m)} = \frac{1}{1+\exp(\beta'_m)}. \end{aligned}$$

Finally, solving for β'_m yields $\beta'_m = \frac{1}{z}\beta_m = \ln(\frac{1}{P} - 1) \quad \rightarrow \quad \beta_m = z \cdot \ln(\frac{1}{P} - 1)$. □

Lemma A.1. *Given any two numbers m, $n \in \mathbb{R}$ such that $m \neq n$, there exists an affine transformation $f_{a,b} : \mathbb{R} \rightarrow \mathbb{R}$ that maps the greater number to* 1 *and the lesser number to* 0.

Proof (Sketch).. Suppose that $m > n$ without loss of generality. We therefore must solve the following system of equations: $f_{a,b}(m) = am + b = 1$ and $f_{a,b}(n) = an + b = 0$. The solution is $a = \frac{-1}{n-m}$ and $b = \frac{m}{n-m} + 1$, which always exists when $m \neq n$. □

B POMCPOW Rollout Policies

ATS solves the HUB-POMDP using *partially observable Monte-Carlo planning with observation widening* (POMCPOW) augmented with a custom rollout policy for estimating the value of leaf nodes in the search tree. We evaluate a *random action* rollout policy, which takes actions uniformly at random from $\mathcal{A} = \mathcal{C} \cup \beta$, a *random arm* rollout policy, which chooses arms uniformly at random from $\mathcal{C}$, and a *best arm* policy, which calculates which arm has the highest expected utility *according to the current belief* b, then always chooses that arm.

Since a utility-maximizing agent will choose arms more often if it believes them to have higher utility, the *best arm* policy rollouts most closely resemble the actions the actual policy would take from belief b, yielding the most accurate value estimates. As a result, ATS with best arm rollouts outperforms the alternatives on the paper recommender domain, as shown in Figure 9. Results are averaged across 25 runs on 20 different paper recommendation tasks.

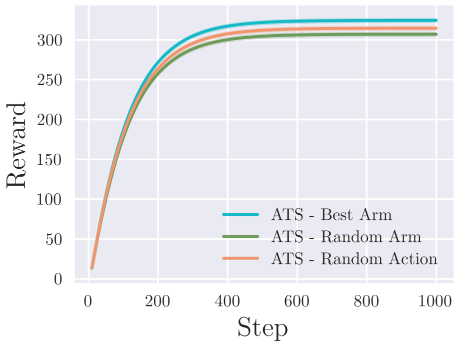
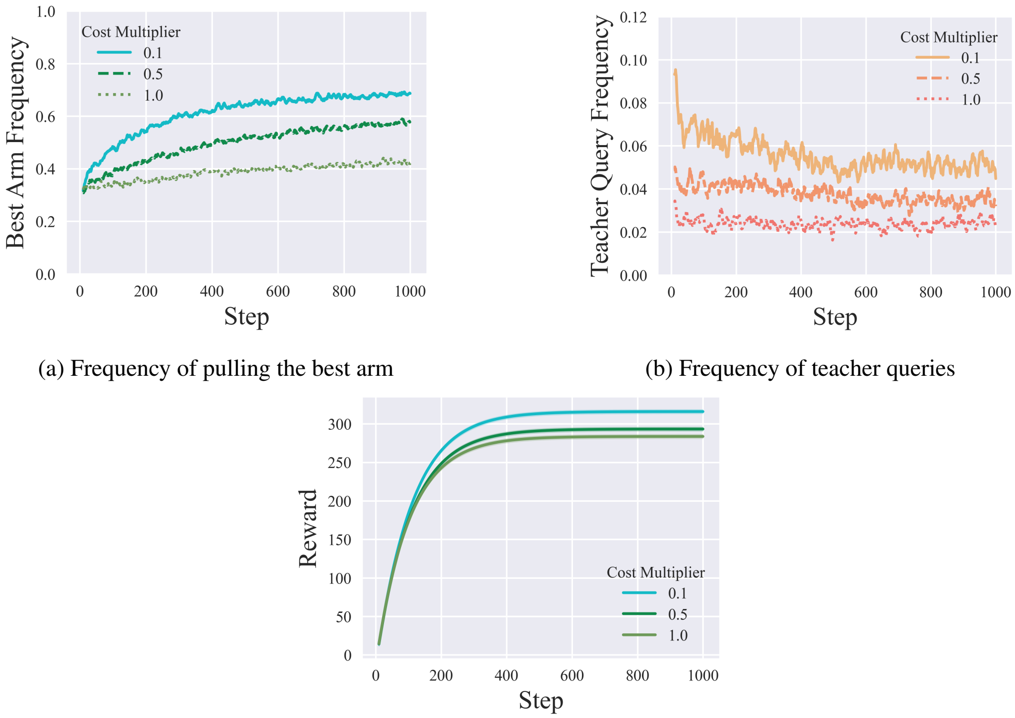


Figure 9: Performance of ATS with various rollout policies. The best arm rollout policy outperforms the random arm and random action rollout policies. All data is averaged across 25 runs on each of 20 HUB problems, smoothed over 10 steps, and discounted with $\gamma = 0.99$.

C HUB Cost effects

We investigate the impacts of teacher query cost on ATS performance by varying professor feedback costs in the paper recommendation domain. We set linear costs $F = \{-1, -2, -3\}$ and scale them by a *cost multiplier*. As in the other paper recommendation experiments, results are averaged across 25 runs on 20 different paper recommendation tasks.

We find that ATS responds rationally to changes in costs, querying teachers more sparingly (Figure 10b) and consequently identifying the best arm more slowly (Figure 10a as overall costs increase. This leads to a slight decrease in overall performance (Figure 10c).



(a) Frequency of pulling the best arm

(b) Frequency of teacher queries

(c) Discounted cumulative reward

Figure 10: ATS behavior and performance varies with teacher query costs. Data is averaged across 25 runs on 20 paper recommendation HUB problems and smoothed over 10 steps.