PAC Apprenticeship Learning with Active IRL

Anonymous authors
Paper under double-blind review

Keywords: inverse reinforcement learning, active learning, imitation learning, Bayesian methods

Summary

As AI systems become increasingly autonomous, aligning their decision-making to human preferences is essential. In domains like autonomous driving or robotics, it is impossible to write down the reward function representing these preferences by hand. Inverse reinforcement learning (IRL) offers a promising approach to infer the unknown reward from demonstrations. However, obtaining human demonstrations can be costly. Active IRL addresses this challenge by strategically selecting the most informative scenarios for human demonstration, reducing the amount of required human effort. As a principled alternative to prior heuristic approaches, we introduce two information-theoretic methods for Active IRL that aim to maximise information about the reward, or alternatively regret, at every step, directly targeting either the *reward learning* or the *apprenticeship learning* objective. We prove that our method yields a probably-approximately-correct (PAC) policy – the first such guarantee for this task. We also illustrate failure modes of prior methods and provide an experimental comparison.

Contribution(s)

1. We formulate two principled information theoretic acquisition functions for active inverse reinforcement learning with Boltzmann rational demonstrations: Reward-EIG and Regret-EIG.
   Context: This gives a more principled alternative to previous, heuristic acquisition functions of Lopes et al. (2009), Brown et al. (2018), and Kweon et al. (2023).
2. For RegretEIG, we prove a lower bound on the expected number of steps of active learning needed to reach a probably-approximately-correct (PAC) apprentice policy.
   Context: This a first such proof for active IRL with an expert that is not perfectly rational. Metelli et al. (2021); Lindner et al. (2022) presented results for the, in many respects much simpler, case of perfectly optimal expert, focusing especially on transfer of a learnt reward to new environment dynamics.

PAC Apprenticeship Learning with Active IRL

Anonymous authors
Paper under double-blind review

Abstract

As AI systems become increasingly autonomous, aligning their decision-making to human preferences is essential. In domains like autonomous driving or robotics, it is impossible to write down the reward function representing these preferences by hand. Inverse reinforcement learning (IRL) offers a promising approach to infer the unknown reward from demonstrations. However, obtaining human demonstrations can be costly. Active IRL addresses this challenge by strategically selecting the most informative scenarios for human demonstration, reducing the amount of required human effort. As a principled alternative to prior heuristic approaches, we introduce two information-theoretic methods for Active IRL that aim to maximise information about the reward, or alternatively regret at every step, directly targeting either the *reward learning* or the *apprenticeship learning* objective. We prove that our method yields a probably-approximately-correct (PAC) policy – the first such guarantee for this task. We also illustrate failure modes of prior methods and provide an experimental comparison.

1 Introduction

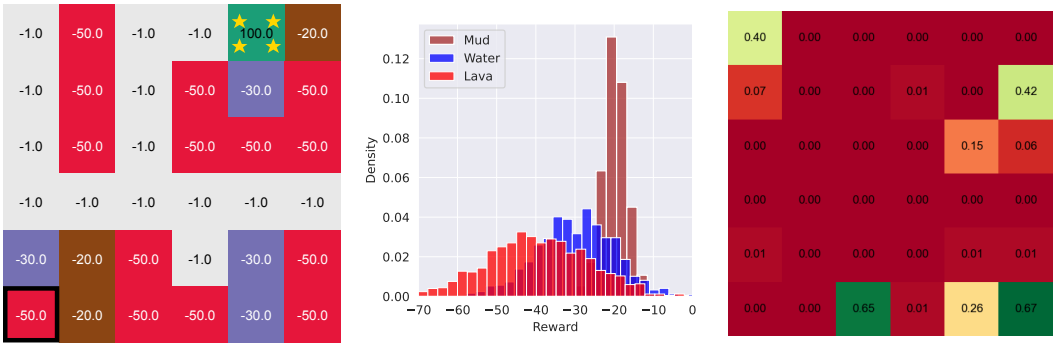
Stuart Russell suggested three principles for the development of beneficial artificial intelligence: its only objective is realizing human preferences, it is initially uncertain about these preferences, and its ultimate source of information about them is human behavior (Russell, 2019). *Apprenticeship learning* via Bayesian *inverse reinforcement learning* (IRL) can be understood as a possible operationalization of these principles: Bayesian IRL starts with a prior distribution over reward functions representing initial uncertainty about human preferences. It then combines this prior with *demonstration* data from a human expert acting approximately optimally with respect to the unknown reward, to produce a posterior distribution over rewards. In apprenticeship learning, this posterior over rewards is then used to produce a policy that should perform well with respect to the unknown reward function.

However, getting human demonstrations requires scarce human time. Also, many risky situations where we would wish AI systems to behave especially reliably may be rare in these demonstration data. Bayesian active learning can help with both by giving queries to a human demonstrator that are likely to bring the most information about the reward.

Most prior methods for active IRL (Lopes et al., 2009; Brown et al., 2018; Metelli et al., 2021) queried the expert for action annotations of particular isolated states. However, in domains such as autonomous driving with a high frequency of actions, it can be much more natural for the human to provide whole trajectories – say, to drive for a while in a simulator – than to annotate a large collection of unrelated snapshots. There is one previous paper on *active IRL with full trajectories* (Kweon et al., 2023) suggesting a heuristic acquisition function whose shortcomings can, however, completely prevent learning, as we will demonstrate. We instead suggest using the principled tools of Bayesian active learning for the task.

The article provides the following contributions:

1. We explain and demonstrate failure modes of existing heuristic methods for active IRL if the goal is to produce a well-performing apprentice policy. In particular, most previous methods are limited



(a) Ground-truth rewards. (b) Current belief over rewards. (c) Reward EIG of each initial state.

Figure 1: Illustration of the active IRL task. (a) shows a gridworld and its true rewards. The lower left corner has a "jail" state with negative reward from which an agent cannot leave. The starred green state is the terminal "goal" state with a large positive reward. The brown, blue, and red states are "mud", "water", and "lava" type states respectively, whose rewards are unknown to the IRL agent. The IRL agent tries to learn the rewards of these three state types from expert demonstrations. (b) shows current distributions over the rewards of the "mud", "water", and "lava" state types respectively, at some particular step of the active learning process. These learned reward distributions are used to calculate an acquisition function (here the reward EIG) of obtaining another expert demonstration starting from each given state, shown in (c). In this case, a demonstration starting in the bottom right state gives the most information about the unknown reward parameters.

to querying for only a single state annotation, as opposed to whole trajectories. Furthermore, we show that the only prior method for whole trajectories can result in repeatedly querying a single uninformative state forever.

2. We propose two acquisition functions based on expected information gain (EIG) – one about the reward, the other about the regret of the apprentice policy.
3. We examine and test 3 possible ways of operationalizing the latter.
4. We provide a theoretical result giving the number of steps in which we can expect regret-EIG to produce a *probably approximately correct* (PAC) apprentice policy – a first such result for expert demonstrations that are not perfectly optimal.
5. We illustrate the performance of our method in a set of gridworld experiments.

2 Task formulation

Let $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, r, \gamma, t_{\max}, \rho)$ be a parameterized Markov decision process (MDP), where $\mathcal{S}$ and $\mathcal{A}$ are finite state and action spaces respectively, $p : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ is the transition function where $\mathcal{P}(\mathcal{S})$ is a set of probability measures over $\mathcal{S}$, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is an (expected) reward function,[1] $\gamma \in (0, 1)$ is a discount rate, $t_{\max} \in \mathbb{N} \cup \{\infty\}$ is the time horizon, and ρ is the initial state distribution.

We assume we are initially uncertain about the reward r, and our initial knowledge is captured by a prior distribution $p(r)$ over rewards, which is a distribution over $\mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ – a space of vectors representing the reward associated with each state-action pair. We also have access to an expert that, given an initial state s_0 of the MDP, can produce a trajectory $\tau_i = \big((s_0^i, a_0^i), \ldots, (s_{n_i}^i, a_{n_i}^i)\big)$, where

[1] Our formulation permits the reward to be stochastic. However, our expert model (1) depends on the rewards only via the optimal Q-function, which in turn depends only on the expected reward. Thus, the demonstrations can only ever give us information about the expectation. Throughout the paper, the learnt reward function can be interpreted either as modeling a deterministic reward, or an expectation of a stochastic reward.

$s_0^i \sim \rho$, $s_{t+1} \sim p(\cdot|s_t, a_t)$, and

$$\pi^E(a_t|s_t) = \frac{\exp(\beta Q^*(s_t, a_t))}{\sum_{a' \in \mathcal{A}} \exp(\beta Q^*(s_t, a'))}\,, \tag{1}$$

which is called a *Boltzmann-rational* policy, given the optimal Q-function Q^* and a hyperparameter β expressing how close to optimal the expert behaviour is (where $\beta = 0$ corresponds to fully random behaviour and $\beta \to +\infty$ would yield the optimal policy).

The task of *Bayesian active inverse reinforcement learning* is to sequentially query the expert to provide demonstrations from initial states $\xi_1, \ldots, \xi_N \in \mathcal{S}$ to gain maximum information about the unknown reward. We start with a (possibly empty) set of expert trajectories $\mathcal{D}_0$ and then, at each step of active learning, we choose an initial state ξ_i for the MDP, from which we get the corresponding expert trajectory τ_i. We then update our demonstration dataset to $\mathcal{D}_i = \mathcal{D}_{i-1} \cup \{\tau_i\}$, and the distribution over rewards to $p(r|\mathcal{D}_i)$, which we again use to select the most informative environment setup ξ_{i+1} in the next step. We repeat until we exhaust our limited demonstration budget N.

This can be done with one of two possible objectives in mind.

The first, which we call the *reward-learning objective*, is relevant when our primary interest is in the reward itself, e.g. when using IRL to understand the motivations of mice in a maze (Ashwood et al., 2022) or the preferences of drivers (Huang et al., 2022). In the active setting, we operationalize this objective as trying to minimize the entropy of the posterior distribution over rewards, once all expert demonstrations have been observed. This is equivalent to maximizing the log likelihood of the true parameter value in expectation, or to maximizing the mutual information between the demonstrations and the reward.

The second objective, which we term the *apprenticeship-learning objective*, uses the final posterior $p(r|\mathcal{D}_N)$ to produce an *apprentice policy*

$$\pi_N^{\mathrm{A}} := \mathrm{argmax}_\pi \mathbb{E}_{r|\mathcal{D}_N}[\mathbb{E}_\tau[\sum_{s_t, a_t \in \tau} \gamma^t r(s_t, a_t)]]\,,$$

where τ is a trajectory with $s_0 \sim \rho$, $s_{t+1} \sim p(\cdot|s_t, a_t)$ and $a_t = \pi_N^{\mathrm{A}}(s_t)$. This approach directly aims to produce a policy which maximises the expected return (and can thus also be understood as a method for imitation learning).

When working with a fixed set of demonstrations in IRL, these objectives are generally closely connected – learning the best possible reward function enables learning a good apprentice policy. However, in the active setting, they can come apart – for instance, once we know a state s has lower reward than s', we may no longer need to make gather further information about rewards in these states for the apprenticeship learning objective as we already know to choose the better one, while the reward-learning objective may motivate further queries to further reduce the uncertainty.

Stemming from a common inspiration in Bayesian active learning, we will present an acquisition function tailored to each of these objectives.

Notation By V_r^π we denote the state-value function of policy π with respect to reward r. V_r^* is then the value function of the optimal policy. A lack of subscript indicates value with respect to the true reward. By $G_r(\tau)$ we denote the return of trajectory τ with respect to r. By $R_r^\pi(s_0)$ we denote the regret of policy π starting from state s_0, i.e. $R_r^\pi(s_0) := V_r^*(s_0) - V_r^\pi(s_0)$, and $R_r^\pi := \mathbb{E}_{s_0 \sim \rho} R_r^\pi(s_0)$. We also call *immediate regret* the quantity $R_{\pi,r}^*(s) = V^*(s) - Q^*(s, \pi(s))$ and also denote by $R_{\pi,r}^*(s, a) = \max\{0, Q^*(s, a) - Q^*(s, \pi(s))\}$ the immediate regret relative to action a in state s.

3 Related work

IRL was first introduced by Russell (1998), preceded by the closely related problem of *inverse optimal control* formulated by Kalman (1964). See Arora & Doshi (2021) and Adams et al. (2022)

for recent reviews of the already extensive literature on IRL. In our work we build upon the Bayesian formulation of the problem introduced by Ramachandran & Amir (2007).

We will now summarize prior work on *active* IRL in particular. We first describe a number of methods which query for single state annotations (which can be cast into our framework from Section 2 as trajectories of length one), and then describe the one previous method which queries for whole trajectories. Lastly, we review a few other works for setups not directly comparable to ours.

3.1 Active learning with single action annotations

The concept of active IRL was first introduced by Lopes et al. (2009). The authors propose an acquisition function equal to the entropy of the posterior predictive distribution about the Boltzmann expert policy, i.e. they query a state maximizing $\alpha_n^{\text{Lopes}}(s) = H(\Pi_s|\mathcal{D}_n)$ where Π_s is the vector of expert action probabilities (according to the posterior predictive distribution).

An issue with this approach is that $H(\Pi_s|\mathcal{D}_n)$ does not take into account the effect of improved knowledge on the apprentice policy. For example, we may know the optimal action in a particular state, but with high uncertainty about the exact action probabilities, while another state may have uncertainty about the optimal action, but lower entropy about exact probabilities of actions. Then, $\alpha_n^{\text{Lopes}}(s)$ would prioritize the latter, which may be suboptimal from the apprenticeship learning perspective. See Appendix A for a full example.

Brown et al. (2018) query the expert by maximising the δ-value-at-risk of the policy loss (i.e. regret) of the current apprentice policy starting from the given initial state, computed as

$$\alpha_n^{\text{Brown}}(s) = \text{VaR}_\delta \left(V^{\pi^*}(s) - V^{\pi^{\text{A}}}(s) | \mathcal{D}_n \right) . \tag{2}$$

This is a risk-aware approach: the states with a high risk of the apprentice action could being much worse than the expert's action are queried. A limitation of this approach is that regret attributed to some initial state s may be due to a choice made further along the trajectory where an expert query would be more informative as shown in Appendix A.

3.2 Active learning with full trajectories

Kweon et al. (2023) query full trajectories with a starting state chosen s_0 to maximise

$$\alpha_n^{\text{Kweon}}(s_0) = \mathbb{E}_{\tau \sim \hat{\pi}_E^{\mathcal{D}_n}} \left[\sum_{s_t \in \tau} \tilde{\alpha}_n(s_t) | s_0 \right] , \tag{3}$$

where

$$\tilde{\alpha}_n(s) := H(\hat{\pi}_E^{\mathcal{D}_n}(a|s)) := \sum_a -\hat{\pi}_E^{\mathcal{D}_n}(a|s) \log \hat{\pi}_E^{\mathcal{D}_n}(a|s),$$

is the entropy of $\hat{\pi}_E^{\mathcal{D}_n}$, the posterior predictive distribution over the expert actions at state s, estimated from demonstration data $\mathcal{D}_n$.

However, note that this action entropy can remain high even in states where we have perfect knowledge but multiple actions are equally good so the Boltzmann rational policy chooses them with equal probabilities, thus resulting in high action entropy. However, querying for extra demonstrations in such states will bring no useful knowledge. In fact, this can result in learning getting completely stuck, sometimes right at the beginning, preventing *any* learning from taking place. We give an example of this in Section 6.

3.3 Other settings

Instead of querying at arbitrary states, Losey & O'Malley (2018) and Lindner et al. (2022) synthesize a policy that explores the environment to produce a trajectory which subsequently gets annotated

by the expert. We instead let the expert produce the trajectory. Buening et al. (2024) query full trajectories in the context of IRL, where the active component arises in the choice of a transition function from a set of transition functions at each step. Büning et al. (2022) also query full trajectories in a different context involving two cooperating autonomous agents. In Sadigh et al. (2017), the expert is asked to provide a relative preference between two sample trajectories synthesized by the algorithm. While this generally provides less information per query than our formulation, it is a useful alternative for situations where providing high-quality demonstrations is difficult for humans.

On the side of theoretical sample complexity of (active) IRL, all prior work assumes a perfectly rational expert policy, which is a stronger assumption than our Boltzmann rationality. In particular, seeing each state once is enough to determine the optimal policy. The first lower bound on the complexity of IRL was given by Komanduru & Honorio (2021) for the case of a β-separable finite set of candidate rewards. Metelli et al. (2021); Lindner et al. (2022); Metelli et al. (2023) then focus on recovering a feasible reward set in settings where also the transition dynamics are only estimated, and address the problem of the transferrability of the learnt reward to environments with different dynamics.

4 Method

Where prior work provided heuristic acquisition functions, we build on the principled approach of Bayesian experimental design (Rainforth et al., 2023), in particular using the expected information gain as our acquisition function. But this raises the question: information about what? The answer depends on our objective and the associated loss function. In Section 2, we presented two such objectives: firstly, learning the reward, which can be operationalized as minimizing the posterior uncertainty represented by the entropy of the posterior; or, secondly, producing an apprentice policy that performs well according to the unknown reward. Only by being clear about what our goal is can we devise an optimal strategy for achieving it. Let us now address how EIG can be used for each of these objectives.

4.1 Reward EIG

In some cases, learning the reward function that an agent is optimizing can be of intrinsic interest. In that case, we can frame active IRL as trying to find a set of queries $\xi_1, \dots, \xi_N$ which minimize, in expectation, our posterior uncertainty about the reward, which can be written as a loss

$$\mathcal{L}_{\text{rew}}(\xi_1, \dots, \xi_N) := H[p(r|\mathcal{D}_N)] \tag{4}$$

where $\mathcal{D}_N$ is a set of expert trajectories sampled from initial states $\xi_1, \dots, \xi_N$. This is equivalent to trying to maximize the posterior probability density of the true reward.

As is usually the case in Bayesian experimental design and optimization, optimizing for the full N-step lookahead is generally intractable, so as a starting point, we consider a greedy formulation where in each step n, we try to optimize the acquisition function

$$\alpha_n^{\text{RewEIG}}(s_0) := \mathbb{E}_{r|\mathcal{D}_n}\left[\mathbb{E}_{\tau|r;\mathcal{D}_n}[\log p(r|\tau;\mathcal{D}_n) - \log p(r|\mathcal{D}_n)]\right] = \\ \mathbb{E}_{r|\mathcal{D}_n}\left[\mathbb{E}_{\tau|r,s_0}[\log p(\tau|r,s_0) - \log p(\tau|s_0;\mathcal{D}_n)]\right],$$

where the expectation over trajectories is taken with respect to ρ, p, and an expert policy that would correspond to the reward r from the outer expectation, taken with respect to the current posterior.

4.2 Regret EIG

What if we are not interested in the reward for its own sake but instead in producing an apprentice agent maximizing the expected posterior reward? While information about the reward is likely to be useful for good apprentice performance, it is only an imperfect proxy.

For example, knowing that certain states are associated with low reward might be sufficient for the apprentice to avoid them. However, learning the exact magnitude of those low rewards would provide no additional benefit to the performance of the apprentice. Despite this, the reward-based EIG would continue to attempt to gather such unnecessary information.

Thus, let us now focus on constructing an acquisition function that tracks the apprenticeship learning objective more closely. *Maximizing* the apprentice return can be framed as *minimizing* the loss

$$\mathcal{L}_{\text{ret}}(\xi_1, \dots, \xi_N) = -\mathbb{E}_{s_0 \sim \rho} \mathbb{E}_{\tau|s_0} G_r(\tau) = -\mathbb{E}_{s_0 \sim \rho} \mathbb{E}_{\tau|s_0} \sum_{s_t, a_t \in \tau} \gamma^t r(s_t, a_t).$$

Minimizing $\mathcal{L}_{\text{ret}}$ is equivalent to minimizing the regret loss

$$\mathcal{L}_{\text{reg}}(\xi_1, \dots, \xi_N) := R_r^{\pi^{\text{A}}{}_N} := \mathbb{E}_{s_0} \mathbb{E}_{\tau|s_0, \pi^{\text{A}}{}_N} \left[V^*(s_0) - G_r(\tau) \right] . \tag{5}$$

However, optimizing for this loss directly is extremely challenging even in the greedy case, since calculating the one-step improvement would involve *maximizing*

$$\alpha_n^{\text{reg}}(s_0) := -\mathbb{E}_{r|\mathcal{D}_n} \mathbb{E}_{\tau_{n+1}|r, s_0} R_r^{\pi^{\text{A}}_{n+1}} . \tag{6}$$

Calculating the regret $R_r^{\pi^{\text{A}}_{n+1}}$ in next step would involve computing the hypothetical updated posterior $p(r|\mathcal{D}_n \cup \{\tau\})$ and then finding an apprentice policy maximizing this posterior, which involves running the Bayesian IRL for each hypothetical expert trajectory, which is extremely costly.

Thus, similar to information-theoretic methods in Bayesian optimization (Wang & Jegelka, 2017) which maximize information about the maximum value rather than directly optimizing the difficult-to-compute improvement in the expected global optimum (the knowledge gradient; Frazier & Powell (2007)), we could instead try to maximize the information gain about the regret, resulting in the acquisition function

$$\alpha^{\text{RegEIG}}(\xi) = \mathbb{E}_{R_r^{\pi^{\text{A}}}} \left[\mathbb{E}_{\tau|R_r^{\pi^{\text{A}}}} [\log p(\tau|R_r^{\pi^{\text{A}}}; \xi) - \log p(\tau|\xi)] \right]. \tag{7}$$

The intuition behind this acquisition function is that learning about the regret will reveal whether and where the current apprentice policy is behaving suboptimally – information that can help us improve the policy.

This seems to be a reasonable heuristic; however, there are situations where it can fail. Suppose that in a particular state, the agent can take action a_0 which they know to yield 0 return. Alternatively, they can take actions a_1 or a_2 and know that one of them yields a reward of +50 and the other -100 but do not know which is which with equal probabilities. Then, the apprentice policy would choose a_0, since it yields higher expected return. They would know for sure that the regret of the policy is 50 when starting from this state. However, since the associated regret distribution is deterministic, the acquisition function would see no information to be gained here and the state would never be queried!

Luckily, there is a way to fix the problem: the regret can be decomposed as $R_r^{\pi^{\text{A}}} = \mathbb{E}_{\tau \sim \rho, \pi^{\text{A}}} \sum_{s_t, a_t \in \tau} \gamma^t \left[V^*(s_t) - Q^*(s_t, \pi^{\text{A}}(s_t)) \right]$ where we call the inner term the *immediate regret* and will henceforth denote it by $R^*_{\pi^{\text{A}}, r}(s)$. Further, we can write $R^*_{\pi^{\text{A}}, r}(s) = \max_a R^*_{\pi^{\text{A}}, r}(s, a)$, where $R^*_{\pi^{\text{A}}, r}(s, a) = \max\{0, Q^*(s, a) - Q^*(s, \pi^{\text{A}}(s))\}$ is the immediate regret of the apprentice policy relative to action a in state s. Then, learning about the values of $R^*_{\pi^{\text{A}}, r}(s, a)$ means learning how good the apprentice action is relative to each other action. Lower-bounding the value by zero means that once we are certain that the apprentice action is optimal, there is no further useful information to be gained.

A further advantage of estimating the expected information gain about $R^*_{\pi^{\text{A}}, r}(s, a)$ is that its distribution in a given state (across all actions) fully determines the distribution of the expert actions, which means that the information gain from observing an expert action in a particular state is fully captured by the information gain about $R^*_{\pi^{\text{A}}, r}(s, \cdot)$ in that given state, which can be used to simplify the EIG calculation.

Thus to summarize, our main acquisition function optimizing for apprentice performance will be EIG about $R^*_{\pi^{\mathrm{A}},r}$, the immediate regret across all states and actions

$$\alpha_n^{\text{IR-EIG}}(s_0) := \mathbb{E}_{R^*_{\pi^{\mathrm{A}},r}|\mathcal{D}_n}\Big[\mathbb{E}_{\tau|R^*_{\pi^{\mathrm{A}},r},s_0}\big[\log p(\tau|s_0, R^*_{\pi^{\mathrm{A}},r}; \mathcal{D}_n) - \log p(\tau|s_0; \mathcal{D}_n)\big]\Big]. \tag{8}$$

5 Producing a PAC Policy

We will now show that the regret EIG acquisition function leads to a probably-approximately-correct (PAC) apprentice policy. In particular, we say that a policy is ϵ, δ-probably-approximately-correct with respect to the current posterior, if, with probability at least $1 - \delta$, its regret is less than ϵ.

We will derive the expected number of steps to reach this condition through the following three steps (the corresponding formal results and their proofs can be found in Appendix B).

1. If the apprentice policy π^{A} is not satisfying the PAC condition, there must exist a state where there is a significant chance that the apprentice policy is making a significantly suboptimal choice relative to the optimal policy – in particular

$$\mathbb{P}_{r|\mathcal{D}_n}\big[V^*_r(s) - Q^*_r(s, \pi^{\mathrm{A}}(s)) \geq (1-\gamma)\epsilon\big] \geq \frac{\delta}{|\mathcal{S}|} \qquad \text{(Lemma B.2).}$$

2. In such a state, there is both a chance that an appropriately constructed apprentice policy is close to optimal and that it is significantly suboptimal (as defined in step 1). Since these two options would result in a sufficiently different expert policies in this state, we can gain a lower-bounded expected amount of information by observing the expert in that state.
3. Since we can keep gaining this minimal amount of information per step as long as the PAC condition is not satisfied and we start with finite entropy about the regret of each policy, eventually, we must satisfy the PAC condition, with an expected number of steps equal to the ratio of the initial entropy and the lower bound on the information gain.

In particular, the last two points translate into the following two theorems:

Theorem 5.1. *For $\epsilon > 0$ and $\delta \in (0, \frac{1}{2}]$, assume that no policy π is (ϵ, δ)-probably-approximately-correct, i.e., $\mathbb{P}[R^\pi_r \geq \epsilon] > \delta$, $\forall \pi$. Then, there exists a state $s \in \mathcal{S}$ such that observing a new expert demonstration at s has an expected information gain of at least*

$$\mathit{EIG}^{\mathit{IR}}_{\mathit{min}}(\epsilon, \delta) = \frac{\delta(1 - e^{-\beta(1-\gamma)\epsilon})^2}{8|\mathcal{A}|^3|\mathcal{S}|}. \tag{9}$$

Then, we can translate this into the following result on the expected number of steps to reach the PAC criterion:

Theorem 5.2. *Let $h_{max} \geq \max_\pi H(p(R^*_{\pi,r}))$ be an upper bound on the entropy (in the sense of a limiting density of discrete points or a suitable discretization) of the joint prior distribution over the state-action immediate regrets of all state-action pairs for any policy. Then, the expected number of steps needed to reach the PAC condition is upper bounded by*

$$h_{max}/\mathit{EIG}_{\mathit{min}}(\epsilon, \delta) = \frac{8h_{max}|\mathcal{A}|^3|\mathcal{S}|}{\delta(1 - e^{-\beta(1-\gamma)\epsilon})^2}. \tag{10}$$

6 Experiments

We evaluated the performance of the two proposed acquisition functions in a set of gridworld experiments with respect to both objectives introduced earlier: the reward learning objective, measured by the entropy of the posterior distribution over rewards, and the apprenticeship learning objective, measured by the regret. We also track the posterior distribution over regrets which directly relates to the PAC criterion.

We evaluate across three types of environment:

1. **Structured gridworld**: Features fewer reward parameters than states. It includes a known goal state with a reward of +100, neutral states with a reward of -1, and three obstacle types with unknown negative rewards with a uniform prior between -100 and 0 independently for each obstacle type. This is was meant as an illustrative example (Figure 1) and a counterexample to the only prior method designed for collecting full trajectories, *action entropy* (Kweon et al., 2023), by including a jail state where all actions are equivalent and which always gets selected by this baseline, thus preventing any useful learning.
2. **10x10 random gridworld with 2 initial states**: Each state has a random reward drawn from the prior, $\mathcal{N}(0, 3)$, with only two possible initial states to test the ability of methods to recognize only relevant parts of the state space. Here we use $\beta = 4$ so the expert behaves closely to optimal.
3. **8x8 random gridworld with fully uniform intial states**: Each state has a random reward drawn from the prior, $\mathcal{N}(0, 3)$, and the initial state distribution is uniform across all states. We use $\beta = 2$ so the expert is fairly stochastic.

We evaluate both the setting where each query results in a full expert trajectory, where we compare against the only prior method (Kweon et al., 2023) as well as random sampling, and the setting where each query results in a single state-action annotation, where we also evaluate against the methods by Lopes et al. (2009) and Brown et al. (2018).

Building on the discussion in the section on regret EIG, we test three versions of computing this. RegEIG – the initial suggestion computing the EIG with respect to the overall regret of the apprentice policy $R_r^{\pi^A}$ (a scalar value, Eq. 6). IR-EIG is the information gain about the immediate regret relative to each other action in each state (Eq. 8). We also tested PAC-EIG, a version of IR-EIG, which instead of focusing on the information gain about the full scalar value, focuses only on a binary random variable indicating whether the regret relative to each other action is $< (1 - \gamma)\epsilon$ for $\epsilon = 0.1$, which is a simplification of IR-EIG focusing on the satisfaction of the PAC criterion for a particular ϵ. The three methods yield roughly similar results so we omit them in some plot for easier legibility.

Each experiment type was run with with 16 different random reward functions, different terminal states (except for the jail environment) and different 2 initial states in the 10x10 environment. The plots display the mean and the standard error across these 16 random instances.

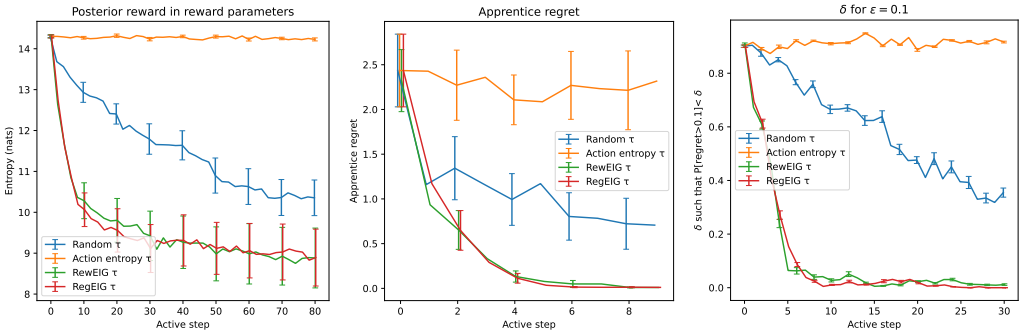
6.1 Results

Results on the simple environment from Figure 1 illustrate a crucial failure mode of the *action entropy* (Kweon et al., 2023) acquisition function – it always queries the jail state and thus fails to learn anything useful, while both reward and regret EIG learn an optimal policy within 10 steps in all 16 instances with similar posterior entropies.

Figure 3 shows the results on the 10x10 gridworld with 2 random initial states and single-state annotations, Figure 4 shows the results for querying single-state annotations on the 8x8 gridworld with a uniform initial state distribution, and Figure 5 results on the 8x8 environment when querying trajectories of maximum length 5.

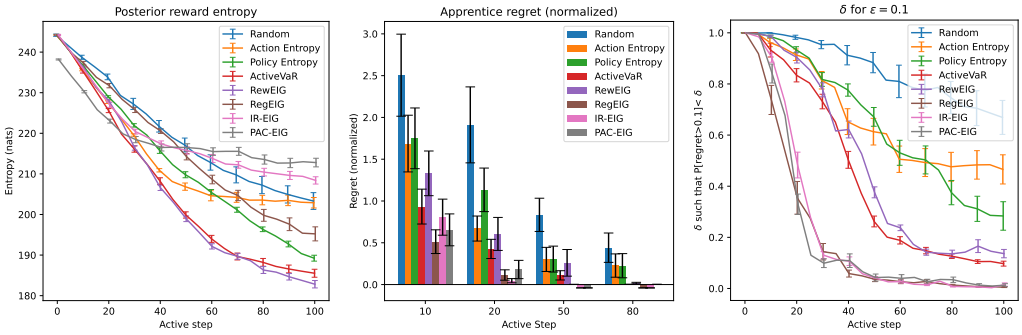
In the case of only 2 initial states on the 10x10 gridworld, we can see our regret-focused acquisition functions to do much better in terms of both actual and posterior regret, reaching a zero true regret, as well as 0.1-0.1-PAC apprentice policy, by step 50. ActiveVaR remains competitive with RewardEIG in terms of the reward-learning objective (posterior entropy), but both fall behind in terms of true and posterior regret.

On the gridworld with fully uniform initial states, we observe that all our information theoretic acquisition functions result in lower posterior reward entropy *and* lower regret than prior methods except for ActiveVaR, which seems to roughly match their performance. Interestingly, the reward-based acquisition function and the regret-based ones seem to perform similarly well on both objectives, suggesting that there is a strong correlation between learning about the reward and learning about the apprentice regret.



(a) Posterior reward entropy (b) True regret (c) Posterior prob. of regret > 0.1

Figure 2: Results of the experiments on the environment with 3 cell types and a jail state with full-trajectory demonstrations.



(a) Posterior reward entropy (b) True regret (c) Posterior prob. of regret > 0.1

Figure 3: Results of the experiments with single state annotations (i.e. $|\tau| = 1$) on the 10x10 fully random gridworld with two initial states. In the barplot (b), results with zero regret are visualized below the horizontal axis to make their presence clearer.

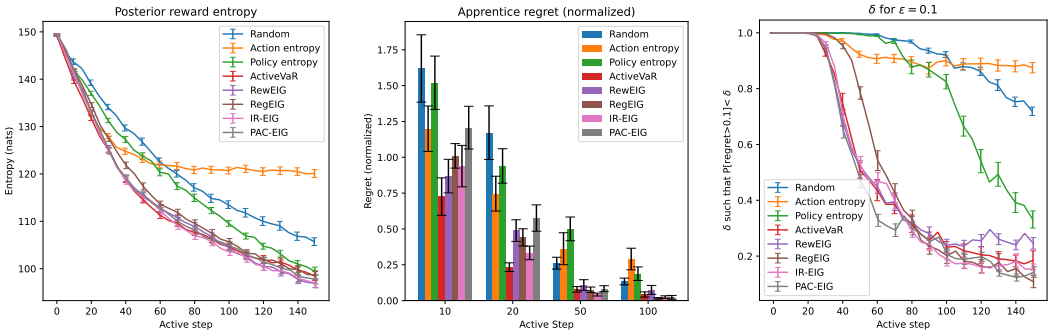
The action entropy acquisition function still stops yielding significant improvements after about step 50 – it again gets stuck querying states that have high action entropy due to multiple actions being similarly good, even if these states do not yield any more information.

7 Discussion and conclusion

In this paper we have proposed new acquisition functions for active IRL, each geared toward one of two possible objectives: learning about an unknown reward function, or producing a well-performing apprentice policy. We have shown that across a set of gridworld experiments, our acquisition functions outperform or at least match prior methods on their respective objective. Furthermore, our immediate-regret EIG acquisition function is a first acquisition function with a regret bound in our setting. While we have so far tested the methods only in finite state spaces, both of them were constructed to generalize also to continuous spaces, which will be addressed in future work.

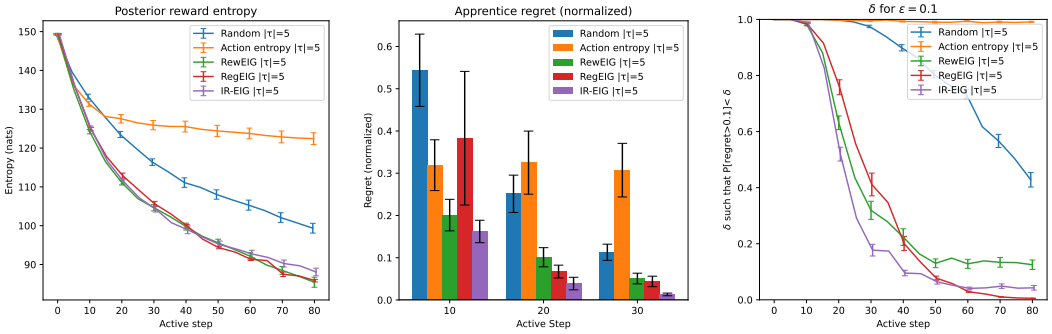
Impact statement

Through this paper, we hope to contribute to more effective and reliable learning of human preferences and values by AI systems, which aims to improve their alignment and facilite their beneficial use.



(a) Posterior reward entropy (b) True regret (c) Posterior prob. of regret > 0.1

Figure 4: Results of the experiments with single state annotations (i.e. $|\tau| = 1$) on the 8x8 fully random gridworld.



(a) Posterior reward entropy (b) True regret (c) Posterior prob. of regret > 0.1

Figure 5: Results of the experiments with expert trajectories of maximum length $|\tau| = 5$ on the 8x8 fully random gridworld.

References

Stephen Adams, Tyler Cody, and Peter A. Beling. A survey of inverse reinforcement learning. *Artificial Intelligence Review*, February 2022. ISSN 0269-2821, 1573-7462. DOI: 10.1007/s10462-021-10108-x. URL https://link.springer.com/10.1007/s10462-021-10108-x.

Saurabh Arora and Prashant Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*, 297:103500, August 2021. ISSN 00043702. DOI: 10.1016/j.artint.2021.103500. URL https://linkinghub.elsevier.com/retrieve/pii/S0004370221000515.

Zoe Ashwood, Aditi Jha, and Jonathan W. Pillow. Dynamic Inverse Reinforcement Learning for Characterizing Animal Behavior. *Advances in Neural Information Processing Systems*, 35:29663–29676, December 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/bf215fa7fe70a38c5e967e59c44a99d0-Abstract-Conference.html.

Ondrej Bajgar, Konstantinos Gatsis, Alessandro Abate, and Michael A. Osborne. Walking the Values in Bayesian Inverse Reinforcement Learning. In *Proceedings of the 40th Conference on Uncertainty in Artificial Intelligence*, 2024.

Daniel S. Brown, Yuchen Cui, and Scott Niekum. Risk-Aware Active Inverse Reinforcement Learning. In *Proceedings of The 2nd Conference on Robot Learning*, pp. 362–372. PMLR, October 2018. URL https://proceedings.mlr.press/v87/brown18a.html. ISSN: 2640-3498.

Thomas Kleine Buening, Victor Villin, and Christos Dimitrakakis. Environment Design for Inverse Reinforcement Learning. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 24808–24828. PMLR, July 2024. URL https://proceedings.mlr.press/v235/kleine-buening24a.html. ISSN: 2640-3498.

Thomas Kleine Büning, Anne-Marie George, and Christos Dimitrakakis. Interactive Inverse Reinforcement Learning for Cooperative Games. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 2393–2413. PMLR, June 2022. URL https://proceedings.mlr.press/v162/buning22a.html. ISSN: 2640-3498.

Alex J Chan and Mihaela van der Schaar. Scalable Bayesian Inverse Reinforcement Learning. *ICLR 2021*, 2021.

Simon Duane, A. D. Kennedy, Brian J. Pendleton, and Duncan Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222, September 1987. DOI: 10.1016/0370-2693(87)91197-X.

Peter Frazier and Warren Powell. The Knowledge Gradient Policy for Offline Learning with Independent Normal Rewards. In *2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, pp. 143–150, Honolulu, HI, USA, April 2007. IEEE. ISBN 978-1-4244-0706-4. DOI: 10.1109/ADPRL.2007.368181. URL http://ieeexplore.ieee.org/document/4220826/.

Matthew D Hoffman and Andrew Gelman. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15:1593–1623, 2014.

Zhiyu Huang, Jingda Wu, and Chen Lv. Driving Behavior Modeling Using Naturalistic Human Driving Data With Inverse Reinforcement Learning. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):10239–10251, August 2022. ISSN 1558-0016. DOI: 10.1109/TITS.2021.3088935. URL https://ieeexplore.ieee.org/abstract/document/9460807. Conference Name: IEEE Transactions on Intelligent Transportation Systems.

R. E. Kalman. When Is a Linear Control System Optimal? *Journal of Basic Engineering*, 86(1): 51–60, March 1964. ISSN 0021-9223. DOI: 10.1115/1.3653115. URL https://doi.org/10.1115/1.3653115.

Abi Komanduru and Jean Honorio. A Lower Bound for the Sample Complexity of Inverse Reinforcement Learning. 2021.

Sehee Kweon, Himchan Hwang, and Frank C. Park. Trajectory-Based Active Inverse Reinforcement Learning for Learning from Demonstration. In *2023 23rd International Conference on Control, Automation and Systems (ICCAS)*, pp. 1807–1812, October 2023. DOI: 10.23919/ICCAS59377.2023.10316798. URL https://ieeexplore.ieee.org/document/10316798. ISSN: 2642-3901.

David Lindner, Andreas Krause, and Giorgia Ramponi. Active Exploration for Inverse Reinforcement Learning. *Advances in Neural Information Processing Systems*, 35:5843–5853, December 2022. URL https://proceedings.neurips.cc/paper/2022/hash/26d01e5ed42d8dcedd6aa0e3e99cffc4-Abstract-Conference.html.

Manuel Lopes, Francisco Melo, and Luis Montesano. Active learning for reward estimation in inverse reinforcement learning. In Wray Buntine, Marko Grobelnik, Dunja Mladenić, and John Shawe-Taylor (eds.), *Machine learning and knowledge discovery in databases*, pp. 31–46, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-642-04174-7.

Dylan P. Losey and Marcia K. O'Malley. Including Uncertainty when Learning from Human Corrections. In *Proceedings of The 2nd Conference on Robot Learning*, pp. 123–132. PMLR, October 2018. URL https://proceedings.mlr.press/v87/losey18a.html. ISSN: 2640-3498.

Alberto Maria Metelli, Giorgia Ramponi, Alessandro Concetti, and Marcello Restelli. Provably Efficient Learning of Transferable Rewards. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 7665–7676. PMLR, July 2021. URL https://proceedings.mlr.press/v139/metelli21a.html. ISSN: 2640-3498.

Alberto Maria Metelli, Filippo Lazzati, and Marcello Restelli. Towards Theoretical Understanding of Inverse Reinforcement Learning. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 24555–24591. PMLR, July 2023. URL https://proceedings.mlr.press/v202/metelli23a.html. ISSN: 2640-3498.

Tom Rainforth, Adam Foster, Desi R. Ivanova, and Freddie Bickford Smith. Modern Bayesian Experimental Design, February 2023. URL http://arxiv.org/abs/2302.14545. arXiv:2302.14545 [cs, stat].

Deepak Ramachandran and Eyal Amir. Bayesian Inverse Reinforcement Learning. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, 2007.

Stuart Russell. Learning agents for uncertain environments (extended abstract). In *Proceedings of the eleventh annual conference on Computational learning theory*, pp. 101–103, Madison Wisconsin USA, July 1998. ACM. ISBN 978-1-58113-057-7. DOI: 10.1145/279943.279964. URL https://dl.acm.org/doi/10.1145/279943.279964.

Stuart Russell. *Human Compatible: Artificial Intelligence and the Problem of Control*. Penguin Random House, 2019.

Dorsa Sadigh, Anca Dragan, Shankar Sastry, and Sanjit Seshia. Active Preference-Based Learning of Reward Functions. In *Robotics: Science and Systems XIII*. Robotics: Science and Systems Foundation, July 2017. ISBN 978-0-9923747-3-0. DOI: 10.15607/RSS.2017.XIII.053. URL http://www.roboticsproceedings.org/rss13/p53.pdf.

Zi Wang and Stefanie Jegelka. Max-value Entropy Search for Efficient Bayesian Optimization. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 12, 2017.

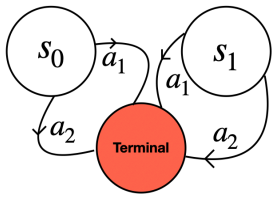


Figure 6: Two-state environment designed to illustrate failure more of Lopes et al. (2009)

A Failure modes of prior methods

For each of the three prior methods for active IRL, we will now present an example of a simple environment where the method makes a clearly suboptimal choice with respect to at least one of the two objectives.

Lopes et al. (2009): Consider an environment with two states $s_{0,1}$ each with two actions $a_{1,2}$ as shown in Figure 6. Suppose through prior knowledge we know that a_1 is optimal in s_1, but we are uncertain about the exact probabilities of a_1, a_2 in this state - and thus $\alpha^{\text{Lopes}}(s_1)$ is high. Furthermore, suppose that in s_0 we lack information about which action is optimal, but the entropy of the probabilities in each state is *lower* $\alpha^{\text{Lopes}}(s_0) < \alpha^{\text{Lopes}}(s_1)$. In this case, budget would be wasted on s_1 rather than learning the optimal action in s_0.

Below we give a concrete example of this effect. We construct a *discrete* prior distribution over rewards which induces a prior over the possible action probabilities in each state. Following the notation of Lopes et al. (2009), denote by $\mu_{s,a}(p)$ the probability that the policy in state s has probability p of taking action a. We construct a prior:

$$\mu_{0,1}(0.4) = 0.3\,, \quad \mu_{0,1}(0.6) = 0.7\,, \tag{11}$$

and reciprocally

$$\mu_{0,2}(0.4) = 0.7\,, \quad \mu_{0,2}(0.6) = 0.3\,, \tag{12}$$

Furthermore, in state s_1 we have

$$\mu_{1,1}(0.7) = 0.5\,, \quad \mu_{1,1}(0.9) = 0.5\,,$$
$$\mu_{1,2}(0.3) = 0.5\,, \quad \mu_{1,2}(0.1) = 0.5\,,$$

With this posterior, the policy is uncertain about whether a_1 or a_2 is optimal in state s_0. In state s_1, although it is unsure of exact probabilities, there is no doubt that a_1 is the optimal action.

The acquisition function α^{Lopes} computes the entropy of these random variables, averaged across actions, in each state:

$$\alpha^{\text{Lopes}}(s_0) = 0.881\,, \quad \alpha^{\text{Lopes}}(s_1) = 1.0\,. \tag{13}$$

Concretely, this acquisition function would query s_1, where the policy *already knows which action is optimal*, rather than s_0 where there is key information to be gained.

(Brown et al., 2018): Consider an environment with two states labelled $s_{0,1}$ and two actions $a_{1,2}$ as shown in Figure 7. Both actions in state s_0 lead to s_1, one with reward $+2$ and the second with -2 (but we do not know which is which). In s_1, both actions lead to a terminal state, and give a reward of -10 and $+10$. Since the potential downside of any policy is maximal at s_0 (-12), the acquisition function would query s_0. On the other hand, querying state s_1 to distinguish the ± 10 rewards would yield a greater reduction in regret.

Consider an intermediate policy which *knows* the absolute values of all the rewards, but not the relative signs: (i.e. $(+2, -2)$ and $(-2, +2)$ are equally likely for $r(a_1|s_0), r(a_2|s_0)$, as are $(+10, -10)$ and $(-10, +10)$ for $r(a_1|s_1), r(a_2|s_1)$. We can easily compute

$$\alpha^{\text{Brown}}(s_0) = 2 + \gamma 10\,, \quad \alpha^{\text{Brown}}(s_1) = 10\,, \tag{14}$$

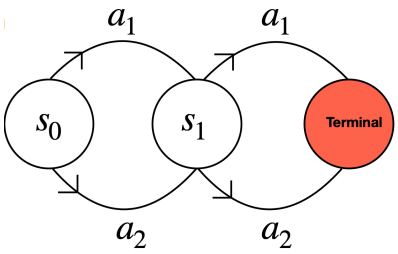


Figure 7: Two state environment to demonstrate failure mode of Brown et al. (2018).

such that for sufficiently large discount factor, state s_0 would be queried by this acquisition function $\text{argmax}_s\{\alpha^{\text{Brown}}(s)\}$. We can compute the reduction in expected regret after querying each of these states. The initial expected total regret for the apprentice policy, averaged over a uniform initial state distribution

$$\mathbb{E}[R] = \frac{1}{2}\big(V^*(s_0) - V^\pi(s_0)\big) + \frac{1}{2}\big(V^*(s_1) - V^\pi(s_1)\big) = \frac{1}{2}(2 + \gamma 10 - 0) + \frac{1}{2}(10 - 0)\,. \quad (15)$$

The expected *total* regret after querying s_0, s_1 respectively:

$$\mathbb{E}[R|s_0] = \frac{1}{2}\big(2{+}\gamma 10{-}2\big){+}\frac{1}{2}\big(10{-}0\big) = (\gamma{+}1)5\,, \quad \mathbb{E}[R|s_1] = \frac{1}{2}\big(2{+}\gamma 10{-}\gamma 10\big){+}\frac{1}{2}\big(10{-}10\big) = 1\,. \quad (16)$$

We therefore observe that whilst Brown et al. (2018) would query s_0, querying s_1 yields a greater reduction in expected regret.

(Kweon et al., 2023): Consider a situation where there is perfect knowledge of the action values in a particular state, but a subset of actions at this state are equivalent (result in the same reward and next state distribution) and tied as optimal. Then the policy will assign uniform probabilities among these actions and due to high action entropy this state will be repeatedly queried. We offer an extreme example of this in Figure 1 which renders this acquisition function useless since it will only ever query the jail cell without gaining any information. Similarly, if in any environment, we allow querying terminal states, the method always queries these since actions have no effect and thus a Boltzmann rational policy would be uniform and thus have maximum entropy.

Uniform random sampling: Consider the single action annotation version of uniform random sampling. Consider a set of $n \times n$ gridworlds with a constant number of states that can yield useful information. A uniform random sampling algorithm will need $\mathcal{O}(n^2)$ steps to visit all of these (so a n grows large, would become unlikely to find a good apprentice policy in any given finite number of step) whereas a method that targets the useful states, such as our methods, would remain $\mathcal{O}(1)$ independently of growing n.

B Theoretical Analysis

We will establish an upper bound on the expected number of expert demonstrations needed to find a policy satisfying our PAC criterion. The proof strategy proceeds in three steps:

1. First, we show that if a policy has positive expected regret, there must exist a state where the policy's action is significantly suboptimal in terms of Q-values.
2. Building on this, we prove that if a policy is not (ϵ, δ)-PAC, then with probability at least δ, there exists a state where the difference between optimal and apprentice policy's Q-values is lower-bounded by a function of ϵ.
3. Finally, we show that in such cases, observing an expert demonstration from an appropriately chosen initial state provides a guaranteed minimum amount of information about the rewards. Since we can only gain a finite amount of information (bounded by the entropy of our prior), this leads to a bound on the number of demonstrations needed.

We begin with our first lemma, which connects policy regret to Q-value differences:

Lemma B.1. *Let π be any policy, r any reward function, and*

$$R_r^\pi = \mathbb{E}_{s_0 \sim \rho_0} \left[V_r^*(s_0) - V_r^\pi(s_0) \right] \geq 0,$$

the regret of that policy. Then there exists a state $s \in \mathcal{S}$ such that

$$Q_r^*(s, \pi^*(s)) - Q_r^*(s, \pi(s)) \geq (1-\gamma) R_r^\pi.$$

Proof. Let us define

$$\Delta_Q = \max_{s \in \mathcal{S}} \left[Q_r^*(s, \pi^*(s)) - Q_r^*(s, \pi(s)) \right].$$

We will prove the lemma by showing that $R_r^\pi \leq \Delta_Q / (1-\gamma)$.

Since $Q_r^\pi(s, \pi(s)) \leq Q_r^*(s, \pi(s))$ (because Q_r^* is the optimal Q-function), we have

$$\begin{aligned} V_r^*(s) - V_r^\pi(s) &= Q_r^*(s, \pi^*(s)) - Q_r^\pi(s, \pi(s)) \\ &\geq Q_r^*(s, \pi^*(s)) - Q_r^*(s, \pi(s)) \\ &\geq 0 \,. \end{aligned}$$

Using the Bellman equation, for any state $s \in \mathcal{S}$ we can write

$$\begin{aligned} V_r^*(s) - V_r^\pi(s) &= Q_r^*(s, \pi^*(s)) - Q_r^\pi(s, \pi(s)) \\ &= Q_r^*(s, \pi^*(s)) - Q_r^*(s, \pi(s)) + Q_r^*(s, \pi(s)) - Q_r^\pi(s, \pi(s)) \\ &\leq \Delta_Q + \big(r(s, \pi(s)) + \gamma \mathbb{E}_{s' | s, \pi(s)} [V_r^*(s')] \big) - \big(r(s, \pi(s)) + \gamma \mathbb{E}_{s' | s, \pi(s)} V_r^\pi(s')] \big) \\ &= \Delta_Q + \gamma \mathbb{E}_{s' | s, \pi(s)} [V_r^*(s') - V_r^\pi(s')] \\ &\leq \Delta_Q + \gamma \max_{s'} [V_r^*(s') - V_r^\pi(s')]. \end{aligned}$$

Here, the first equality just replaces state values by the corresponding Q-values, the second line adds and subtracts the same term, the third line uses the definition of Δ_Q for the first term and expands the latter two Q-values using the Bellman equation, the third just cancels out the repeated reward term. The final inequality follows because the expectation over next states is bounded by the maximum.

Since this inequality holds for all $s \in \mathcal{S}$, it holds also for the state maximizing the left-hand side, so we get

$$\max_s [V_r^*(s) - V_r^\pi(s)] \leq \Delta_Q + \gamma \max_{s'} [V_r^*(s') - V_r^\pi(s')] \tag{17}$$

which can be readily rearranged into

$$\max_s [V_r^*(s) - V_r^\pi(s)] \leq \frac{\Delta_Q}{1-\gamma}.$$

Thus

$$R_r^\pi = \mathbb{E}_{s_0 \sim \rho_0} \left[V_r^*(s_0) - V_r^\pi(s_0) \right] \leq \max_s [V_r^*(s) - V_r^\pi(s)] \tag{18}$$

$$\leq \frac{\Delta_Q}{1-\gamma} = \frac{1}{1-\gamma} \max_{s \in \mathcal{S}} \left[Q_r^*(s, \pi^*(s)) - Q_r^*(s, \pi(s)) \right], \tag{19}$$

which completes the proof. □

Lemma B.2. *Let π be the apprentice policy at step n. For any $\delta \in (0, \frac{1}{2}]$, let $R_{n,\delta}^\pi$ be the $(1-\delta)$-quantile of the regret distribution with respect to the current posterior distribution over rewards, i.e., $R_{n,\delta}^\pi$ satisfies*

$$\mathbb{P}_{r | \mathcal{D}_n} (R_r^\pi \geq R_{n,\delta}^\pi) = \delta.$$

Then, there exists a state $s \in \mathcal{S}$ such that

$$\mathbb{P}_{r | \mathcal{D}_n} \left(Q_r^*(s, \pi_r^*(s)) - Q_r^*(s, \pi(s)) \geq (1-\gamma) R_{n,\delta}^\pi \right) \geq \frac{\delta}{|\mathcal{S}|}.$$

Proof. Let us define the set of reward functions under which π has high regret:

$$\mathcal{H} = \left\{ r : R_r^\pi \geq R_{n,\delta}^\pi \right\}.$$

By definition of the quantile $R_{n,\delta}^\pi$, we have

$$\mathbb{P}_{r|\mathcal{D}_n}(r \in \mathcal{H}) = \delta.$$

For each $r \in \mathcal{H}$, applying Lemma B.1, we know there exists a state $s_r \in \mathcal{S}$ such that

$$Q_r^*(s_r, \pi_r^*(s_r)) - Q_r^*(s_r, \pi(s_r)) \geq (1-\gamma) R_r^\pi \geq (1-\gamma) R_{n,\delta}^\pi.$$

Now, consider the collection of states $\{s_r : r \in \mathcal{H}\}$. Since the state space $\mathcal{S}$ is finite with cardinality $|\mathcal{S}|$, by the pigeonhole principle, there must exist at least one state $s \in \mathcal{S}$ such that

$$\mathbb{P}_{r|\mathcal{D}_n}\left(r \in \mathcal{H} \text{ and } s_r = s\right) \geq \frac{\delta}{|\mathcal{S}|}.$$

For this state s, whenever $r \in \mathcal{H}$ and $s_r = s$, we have

$$Q_r^*(s, \pi_r^*(s)) - Q_r^*(s, \pi(s)) \geq (1-\gamma) R_{n,\delta}^\pi.$$

Therefore,

$$\mathbb{P}_{r|\mathcal{D}_n}\left(Q_r^*(s, \pi_r^*(s)) - Q_r^*(s, \pi(s)) \geq (1-\gamma) R_{n,\delta}^\pi\right) \geq \frac{\delta}{|\mathcal{S}|},$$

which completes the proof. □

This lemma extends our previous result to the probabilistic setting of Bayesian IRL. While Lemma B.1 showed that high regret implies the existence of a state with poor action choice, this lemma shows that if our policy has a significant probability of high regret, there must be at least one state where it has a significant probability of making a poor action choice.

Theorem 5.1. *For $\epsilon > 0$ and $\delta \in (0, \frac{1}{2}]$, assume that no policy π is (ϵ, δ)-probably-approximately-correct, i.e., $\mathbb{P}[R_r^\pi \geq \epsilon] > \delta,\ \forall \pi$. Then, there exists a state $s \in \mathcal{S}$ such that observing a new expert demonstration at s has an expected information gain of at least*

$$EIG_{min}^{IR}(\epsilon, \delta) = \frac{\delta(1 - e^{-\beta(1-\gamma)\epsilon})^2}{8|\mathcal{A}|^3|\mathcal{S}|}. \tag{9}$$

Proof. We will prove the theorem by showing that under its assumptions, there exists a state s and an alternative action $a^* \neq \pi^{\mathrm{A}}(s)$ that has a chance of being significantly better than the apprentice action. If it is significantly better, it is significantly more likely to get selected by the expert than if it is inferior to $\pi^{\mathrm{A}}(s)$. Since the observation distributions in the two cases are different, this allows us to put a lower bound on the expected information gained by observing the expert.

Under the assumptions of this theorem and using Lemma B.2, there exists a state such that

$$\mathbb{P}_{r|\mathcal{D}_n}\left[Q_r^*(s, \pi_r^*(s)) - Q_r^*(s, \pi^{\mathrm{A}}(s)) \geq (1-\gamma)\epsilon\right] \geq \frac{\delta}{|\mathcal{S}|}.$$

Let us denote by E the event $\left\{Q_r^*(s, \pi_r^*(s)) - Q_r^*(s, \pi^{\mathrm{A}}(s)) \geq (1-\gamma)\epsilon\right\}$. This event can be partitioned into events $E_a = E \cap \{a = \pi_r^*(s)\}$ for $a \in \mathcal{A} \setminus \{\pi^{\mathrm{A}}(s)\}$. Let a^* be the action whose partition $E^* := E_{a^*}$ has the highest probability under the posterior. Then $\mathbb{P}[E^*] \geq \frac{\delta}{(|\mathcal{A}|-1)|\mathcal{S}|}$.

Let E_A be the event of the apprentice action $\pi^{\mathrm{A}}(s)$ being optimal, i.e.,

$$E_A = \{\pi^{\mathrm{A}}(s) = \pi_r^*(s)\} = \{Q_r^*(s, a) - Q_r^*(s, \pi^{\mathrm{A}}(s)) \leq 0,\ \forall a \in \mathcal{A}\}.$$

Since π^{A} is defined as the policy maximizing the probability of being optimal, we have $\mathbb{P}[E_A] \geq \frac{1}{|\mathcal{A}|}$. Note that this event is disjoint from E^*. For completeness, we define E_C as the complement of $E^* \cup E_A$, so E_A, E^*, and E_C form a partition of the event space.

Now, if we denote by A the action taken by the expert in state s seen as a random variable, we can decompose the mutual information between A and E as

$$I(A;E) = \sum_{E \in \{E^*, E_A, E_C\}} \mathbb{P}[E]\; D_{\mathrm{KL}}(p(A|E) \| p(A)).$$

To finish the proof, we need to put a lower bound on this mutual information. We have already given lower bounds on $\mathbb{P}[E^*]$ and $\mathbb{P}[E_A]$. We will now provide a lower bound on the corresponding KL terms by first lower-bounding the total variation distance between $p(A|E^*)$ and $p(A|E_A)$.

In the event E_A, we have $Q_r^*(s, \pi^{\mathrm{A}}(s)) \geq Q_r^*(s, a^*)$, so under the Boltzmann-rational policy, we have $p(\pi^{\mathrm{A}}(s)|s; E_A) \geq p(a^*|s; E_A)$.

On the other hand, in case of E^*, we have

$$\begin{aligned} p(\pi^{\mathrm{A}}(s)|s; E^*) &= \frac{1}{Z} e^{\beta Q_r^*(s, \pi^{\mathrm{A}}(s))} \leq \frac{1}{Z} e^{\beta (Q_r^*(s,a^*) - (1-\gamma)\epsilon)} \\ &= e^{-\beta(1-\gamma)\epsilon} p(a^*|s; E^*) \\ &= p(a^*|s; E^*) - (1 - e^{-\beta(1-\gamma)\epsilon}) p(a^*|s; E^*). \end{aligned}$$

Since under E^*, a^* is the optimal action, it is also the most likely action under the Boltzmann expert policy, so we have $p(a^*|s; E^*) > \frac{1}{|\mathcal{A}|}$, which gives us

$$p(a^*|s; E^*) - p(\pi^{\mathrm{A}}(s)|s; E^*) > (1 - e^{-\beta(1-\gamma)\epsilon}) \frac{1}{|\mathcal{A}|}. \tag{20}$$

We can use this to bound the total variation distance:

$$\begin{aligned} D_{\mathrm{TV}}^{\mathrm{total}} &= D_{\mathrm{TV}}(p(A|E_A), p(A)) + D_{\mathrm{TV}}(p(A|E^*), p(A)) \\ &\geq D_{\mathrm{TV}}(p(A|E_A), p(A|E^*)) \\ &\geq \frac{1}{2} \big(p(\pi^{\mathrm{A}}(s)|E_A) - p(\pi^{\mathrm{A}}(s)|E^*) + p(a^*|E^*) - p(a^*|E_A) \big) \\ &\geq \frac{1}{2} \big(p(a^*|E^*) - p(\pi^{\mathrm{A}}(s)|E^*) \big) \\ &> \frac{1}{2} (1 - e^{-\beta(1-\gamma)\epsilon}) \frac{1}{|\mathcal{A}|} \end{aligned}$$

where we used the triangle inequality in the first step, the definition of total variation distance in the second step (omitting non-negative terms corresponding to actions other than π^{A} and a^*), the fact that $p(\pi^{\mathrm{A}}(s)|E_A) \geq p(a^*|E_A)$ in the third step, and plugging in Equation (20) in the final step.

Since both TV terms on the left-hand side are non-negative, we have

$$\begin{aligned} \max\{ & D_{\mathrm{TV}}(p(A|E_A), p(A)), \\ & D_{\mathrm{TV}}(p(A|E^*), p(A)) \} \\ & > \frac{1}{4|\mathcal{A}|} (1 - e^{-\beta(1-\gamma)\epsilon}). \end{aligned}$$

Applying Pinsker's inequality, this gives us

$$\begin{aligned} \max\{ & D_{\mathrm{KL}}(p(A|E_A) \| p(A)), \\ & D_{\mathrm{KL}}(p(A|E^*) \| p(A)) \} \\ & > \frac{1}{8|\mathcal{A}|^2} (1 - e^{-\beta(1-\gamma)\epsilon})^2. \end{aligned}$$

This finally allows us to establish that

$$\begin{aligned} I(A;E) &= \sum_{E \in \{E^*, E_A, E_C\}} \mathbb{P}[E] D_{\mathrm{KL}}(p(A|E) \| p(A)) \\ &\geq \min\{\mathbb{P}[E_A], \mathbb{P}[E^*]\} \\ &\quad \times \max\{D_{\mathrm{KL}}(p(A|E_A) \| p(A)), D_{\mathrm{KL}}(p(A|E^*) \| p(A))\} \\ &> \min\left\{\frac{1}{|\mathcal{A}|}, \frac{\delta}{(|\mathcal{A}|-1)|\mathcal{S}|}\right\} \\ &\quad \times \frac{1}{8|\mathcal{A}|^2}(1 - e^{-\beta(1-\gamma)\epsilon})^2 \\ &= \frac{\delta}{8|\mathcal{A}|^3|\mathcal{S}|}(1 - e^{-\beta(1-\gamma)\epsilon})^2. \end{aligned}$$

The first inequality follows from the fact that all three terms in the sum are non-negative. The second inequality just plugs in results previously derived in this proof. The final step resolves the minimum as its second term using the assumption that $\delta \leq \frac{1}{2}$ and $|\mathcal{A}| \geq 2$.

Since the distribution over events E is fully defined in terms of the distribution over Q-values, this mutual information between A and E, which is also the expected information gain about E from observing the expert action, is a lower bound on the information gain about the Q-values (and thus also the rewards). This completes the proof. □

B.1 Information-Theoretic Bounds

To establish bounds on the number of steps needed, we first define suitable discrete random variables that capture the information relevant to achieving the PAC criterion. For any state $s \in \mathcal{S}$ and action $a \in \mathcal{A}$, we define

$$R^*_{s,a} := Q^*_r(s, \pi^*_r(s)) - Q^*_r(s, a) \tag{21}$$

to be the random variable capturing the immediate regret of action a in state s. We then define $E_{s,a}$ as a ternary random variable that categorizes this regret:

$$E_{s,a} = \begin{cases} \text{“zero”} & \text{if } R^*_{s,a} = 0 \\ \text{“small”} & \text{if } 0 < R^*_{s,a} \leq \epsilon \\ \text{“large”} & \text{if } R^*_{s,a} > \epsilon \end{cases} \tag{22}$$

We aggregate this information across all actions in state s by defining

$$E_s := (E_{s,a})_{a \in \mathcal{A}},$$

which is a vector of $|\mathcal{A}|$ ternary variables. Finally, we define

$$E := (E_s)_{s \in \mathcal{S}} \tag{23}$$

to be a random variable composed of the variables E_s for all states.

These discrete random variables allow us to work with standard (non-differential) entropy, avoiding some of the technical challenges of continuous distributions while still capturing the information needed to establish PAC bounds. In particular, knowing E with certainty would tell us which actions are optimal in each state.

Theorem 5.2. *Let $h_{max} \geq \max_\pi H(p(R^*_{\pi,r}))$ be an upper bound on the entropy (in the sense of a limiting density of discrete points or a suitable discretization) of the joint prior distribution over the state-action immediate regrets of all state-action pairs for any policy. Then, the expected number of steps needed to reach the PAC condition is upper bounded by*

$$h_{max}/EIG_{min}(\epsilon, \delta) = \frac{8h_{max}|\mathcal{A}|^3|\mathcal{S}|}{\delta(1 - e^{-\beta(1-\gamma)\epsilon})^2}. \tag{10}$$

Proof. The minimal expected information gain guaranteed by Theorem 5.1 is fully derived from components of the random variable E. Thus, at every step of active learning where we have not yet achieved the PAC criterion, we can gain at least $\text{EIG}_{\text{min}}(\epsilon, \delta)$ information about E.

Let H_n denote the entropy of E after n steps of active learning. By the properties of entropy and information gain:

1. $H_n \geq 0$ for all n (non-negativity of entropy)
2. $H_0 = h_{\text{prior}}$ (initial entropy)
3. $H_{n+1} \leq H_n - \text{EIG}_{\text{min}}(\epsilon, \delta)$ for all n where the PAC criterion is not met (guaranteed information gain)

Let N be the number of steps needed to reach the PAC criterion. Then:

$$\begin{aligned} 0 &\leq H_N \\ &\leq H_0 - N \cdot \text{EIG}_{\text{min}}(\epsilon, \delta) \\ &= h_{\text{prior}} - N \cdot \text{EIG}_{\text{min}}(\epsilon, \delta) \end{aligned}$$

Solving for N:

$$N \leq \frac{h_{\text{prior}}}{\text{EIG}_{\text{min}}(\epsilon, \delta)} \tag{24}$$

The result follows by substituting the expression for $\text{EIG}_{\text{min}}(\epsilon, \delta)$ from Theorem 5.1. □

Corollary B.3. *For any prior distribution over rewards, the expected number of steps to reach the PAC condition is at most*

$$\log(3)|\mathcal{S}||\mathcal{A}|/EIG_{min}(\epsilon, \delta) = \frac{8\log(3)|\mathcal{A}|^4|\mathcal{S}|^2}{\delta(1 - e^{-\beta(1-\gamma)\epsilon})^2}. \tag{25}$$

Proof. The random variable E aggregates $|\mathcal{S}||\mathcal{A}|$ ternary random variables, so it can take at most $3^{|\mathcal{S}||\mathcal{A}|}$ values. Thus, its maximum entropy is $-\log(1/3^{|\mathcal{S}||\mathcal{A}|}) = |\mathcal{S}||\mathcal{A}|\log(3)$. The result follows by plugging this maximum entropy into Theorem 5.2. □

While we do not claim this bound is tight, it provides a useful characterization of how the sample complexity scales with the problem parameters. In particular, it shows polynomial dependence on the size of the state and action spaces, and inverse dependence on both the allowed suboptimality ϵ and failure probability δ.

B.2 Notes on possible improvements

B.2.1 Tighter bound for large state spaces

Note that the bound from Lemma B.2 can be tightened if the state space is large and only a subset is reachable within an effective horizon. In that case $|S|$ can be replaced by the number of states reachable from the initial states within $1/(1-\gamma)$ steps.

B.2.2 Static policy

The bound includes the entropy of each action in each state. In fact, it may be enough to focus on a single action in each state, since we want to identify only a particular PAC policy, rather than reducing entropy of all of the components of $E(s)$ in every state. This should allow us to exclude a factor of $|A|$ from the bound on the expected number of steps to reach the PAC condition.

B.2.3 Generalization of Theorem 5.1

Note that the proof of Theorem 5.1 can be adapted to work with any policy that has a high probability of taking a low-regret action, e.g. policies with

$$\mathbb{P}[R^*_\pi(s,a) < \frac{\epsilon}{2}] > \frac{1}{|\mathcal{A}|}. \tag{26}$$

In practice, the various notions of the "best" apprentice policy tend to correlate, so the lowest-expected-regret policy is likely to satisfy this assumption, and if not, it can be adjusted by moving additional necessary probability weight onto the most-likely-optimal action, in which case this active learning strategy can guarantee the PAC condition for the lowest-expected-regret policy, which would usually be the default one to use in practice.

C Resulting acquisition function and its computation

A practically useful acquisition function should account for one more important point: in order to get a policy with low expected regret, we do not need to reduce the expected immediate regret of all points, but just those that are likely to get visited by the apprentice policy π^{A}. Let $\nu_A(s) := \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \mathbb{I}[s_t = s]]]$ be the discounted expected occupancy of state s. Then, we can set

$$\tilde{\alpha}(s) := \nu_A(s) I(A_s, E_s). \tag{27}$$

to be the acquisition function for querying single states.

However, we also wish to have an acquisition function for collecting full trajectories. A naive approach employed by some prior work (Kweon et al., 2023) would be summing the individual

$$\alpha'_n(s_0) := \mathbb{E}_{r|\mathcal{D}_n}\left[\mathbb{E}_{\tau|r}\left[\sum_{s\in\tau} \tilde{\alpha}(s)\right]\right] \tag{28}$$

However, the sum in Eq. (28) neglects correlation between the regrets in different states (and, worse, autocorrelation if a state is visited multiple times). We can instead estimate the full expected information gain about our variable E from the new expert trajectory. Then, dropping the weighing for the moment,

$$\mathrm{EIG}_E(s_0) = \mathbb{E}_{\tau,E}\big[\log p(\tau|E) - \log p(\tau)\big]. \tag{29}$$

We assume that the Bayesian IRL method we use to estimate this is able to give us samples from the current posterior over Q-values. Given a Q-value and an initial state, we can sample the corresponding hypothetical expert trajectories $\tau|Q$. Also, E is a cheaply-computable function of Q, so we can also easily convert the samples of Q into samples of E. Then, the only remaining challenge in computing EIG_E is estimating $p(\tau|E)$.

If the state space is small and we have a lot of Q samples, we can estimate $p(\tau|E) = \frac{1}{|\mathcal{Q}_E|}\sum_{Q\in\mathcal{Q}_E} p(\tau|Q)$. However, note that for a given policy, there are $3^{|S|}$ possible values of E^π, so even for a moderate size of the state space, the number of Q corresponding to each E could be small. At the same time, the components of E^π corresponding to states far away from the trajectory are unlikely to share much mutual information with the trajectory. Thus we suggest using the bound

$$I(\tau, E) \geq I(\tau, E^\pi_\tau) \tag{30}$$

where $E^\pi_\tau := \big(E_s\big)_{s\in\mathcal{S}_\tau}$ for $\mathcal{S}_\tau$ being some neighbourhood of τ in the state space, including all states on the trajectory τ plus states that can be quickly reached from the trajectory.

To again add the weighing, we can note that since the transition probabilities are the same for both components,

$$\log p(\tau|E) - \log p(\tau) = \sum_{s,a\in\tau} \log p(a|s;E) - \log p(\tau|s) \tag{31}$$

which naturally allows us to re-introduce the weights as

$$\sum_{s,a\in\tau} \nu_\pi(s,a)(\log p(a|s;E) - \log p(\tau|s)) \tag{32}$$

resulting in an acquisition function

$$\mathrm{EIG}_E(s_0) = \mathbb{E}_{\tau,E}\big[\sum_{s,a\in\tau} \nu_\pi(s,a)(\log p(a|s;E_\tau) - \log p(\tau|s))\big]. \tag{33}$$

D Experiment details

D.1 Basic parameter values

In the three environments (structured 6x6, random 8x8, and random 10x10) we used $\beta = 4, 2, 4$ respectively, $\gamma = 0.9$, and an infinite horizon (but all environments contained terminal states). We started with an empty set of demonstrations (implemented as a single, uninformative observation of a dummy sink state) and then ran active learning for 150 steps.

For ActiveVaR, we used $\delta = 0.05$ (same as the original paper). For policy entropy, we used the entropy of the discretized distribution for each action (as proposed by the authors) with K=10 buckets. For our regret-based acquisition functions, we use regret discretization with $\epsilon = 0.1$ for the full regret and $(1-\gamma)\epsilon = 0.01$ for the immediate regret.

D.2 Environments

The gridworld environments have 5 actions, corresponding to staying in place and moving in the four directions. Furthermore, there is a probability of 0.1 of random action being executed instead of the intended one. If an action would result in crossing the edge, the agent instead remains in place. The gridworlds use a state-only reward (awarded upon executing any action in the given state).

The 8x8, and 10x10 fully random environments were generated as follows:

1. Each state was assigned a random reward drawn independently from $\mathcal{N}(0, 3)$ (i.e. mostly yielding rewards between -10 and 10).
2. Each state was then marked as terminal with an independent probability of 0.1.
3. The top 10% of states with highest reward were further marked as terminal (producing terminal goal states, which may, however, sometimes be avoided by the optimal policy in favour of staying forever in other positive states).
4. The initial state distribution is either uniform across the whole state space, or, in the case of the 10x10 gridworld, 2 non-terminal initial states were chosen randomly uniformly.[2]

D.3 Bayesian IRL methods

Our active learning uses a Bayesian IRL method as a key component. In our experiments, we used two methods based on Markov chain Monte Carlo (MCMC) sampling: on the structured environment, we used PolicyWalk (Ramachandran & Amir, 2007), while on the environment with a different random reward in every state, we used the faster ValueWalk (Bajgar et al., 2024), which performs the sampling primarily in the space of Q-functions before converting into rewards. We also tried a method based on variational inference (Chan & van der Schaar, 2021), but we found its uncertainty estimates unreliable for the purposes of active learning.

For MCMC sampling, we used Hamiltonian Monte Carlo (Duane et al., 1987) with the no-U-turns (NUTS) sampler (Hoffman & Gelman, 2014) and automatic step size selection during warm-up

[2]Note that the implementation allows the two initial states to collide, producing only a single initial state in $1/81$ of the cases, but this was not the case for any of our 16 random seeds.

(starting with a step size of 0.1). At every step of active learning, we ran the MCMC sampling from scratch using all demonstrations available up to that point. We ran for 100 warm-up steps and then 200, 500, and 1000 on the three environments respectively. For subsequent usage, we use every other sample to reduce autocorrelation.

D.4 Metrics

On the first two environments, we use KNN entropy estimation to calculate posterior entropy with K=5. This method is known to struggle in high dimensions, which we also observed in the case of the 10x10 gridworld (which has a 100-dimensional reward space), so there, we estimate the entropy by the entropy of a multivariate normal distribution with the mean and covariance matrix estimated from the MCMC samples.

Regret was calculated relative to the expected return of the optimal policy, calculated using value iteration with a tolerance of 1e-5. Posterior regret samples were similarly calculated relative to the optimal return with respect to each of the posterior reward samples (which were calculated using the optimal Q-value samples which get produced by the Bayesian IRL methods).

When aggregating true regret across environment instances, we also normalized the regret for each random environment instance by the average regret across all methods across the first 32 steps of active learning to account for the possibly different scales and different learning difficulties of the random environments.

D.5 Implementation

The experiments were implemented using Python 3.10, PyTorch 2.5.1, and Pyro 1.8.6. We will publish our full code for both the experiments and the associated result analysis on Github once the anonymity requirement is lifted.

D.6 Timing

The computational time per step of active IRL is dominated by the time necessary to collect the Bayesian IRL MCMC samples, which ranges between 5 seconds for the 100+200 samples on the structured gridworld to about 5 minutes for the 100+1000 samples on the 10x10 gridworld in a single CPU thread. The overhead of all acquisition functions on top of that is below 0.03 and can thus be considered negligible.

Reproducing all our experiments thus takes less than a day on a CPU with 128 threads (we used AMD Ryzen Threadripper 3990X at 2.2GHz).