AMBER: An Entropy Maximizing Environment Design Algorithm for Inverse Reinforcement Learning

Paul Nitschke [1] **Lars L. Ankile** [1] **Eura Shin** [1] **Siddharth Swaroop** [1] **Finale Doshi-Velez** [1] **Weiwei Pan** [1]

Abstract

In Inverse Reinforcement Learning (IRL), we learn the underlying reward function of humans from observations. Recent work shows that we can learn the reward function more accurately by observing the human in multiple related environments, but efficiently finding informative environments is an open question. We present AMBER, an information-theoretic algorithm that generates highly informative environments. With theoretical and empirical analysis, we show that AMBER efficiently finds informative environments and improves reward learning.

1. Introduction

When studying sequential human decision making, we often model the human as a Reinforcement Learning (RL) agent. In Inverse Reinforcement Learning (IRL) we aim to learn the human's *reward function*, i.e. how the human values various outcomes of a task, from their behavior. By interpreting the reward function, we can form hypotheses about the *reasons* behind their behavior. For example, Yu et al. (2019) use IRL to determine how doctors decide the amount of sedative dosing in the intensive care unit. Thereby, we can better understand the decision making process of successful doctors. While promising, traditional IRL approaches face a common and significant challenge, *reward non-identifability*: many different reward functions can induce the same policy, i.e, one may not be able to learn the *ground-truth* reward function (Cao et al., 2021; Kim et al., 2021; Metelli et al., 2021; 2023).

In a recent work, *Environment Design for IRL*, Buening et al. (2024) introduces an algorithm for improving reward identifiability by iteratively observing the human in a set of related environments. In each iteration, the Environment-Design algorithm observes the human interact in a new environment. Each new environment is designed to increase our knowledge about the unknown reward function. While promising, the algorithm is computationally expensive and does not have theoretical performance guarantees.

In this paper, we develop a theoretically-grounded and efficient Environment-Design approach to mitigate the non-identifiability of reward learning in IRL. Specifically, our contributions are: (1) we provide intuition and theory to characterize high information-gain environments, (2) we efficiently compute these environments during each iteration of our algorithm, and (3) we provide theoretical and empirical analysis of the performance of our algorithm, showing that our algorithm outperforms existing baselines for reward learning. Throughout this work, we draw parallels to ideas from neuroscience and psychology.

We start with Related Works and Background, introducing key ideas such as IRL and Behavior Maps (Ankile et al., 2023). We set up the environment design problem setting in section 4. In section 5 we show how we can find environments with high information gain by quantifying the entropy of their Behavior Maps. This motivates AMBER, **A**ctive **M**aximization of **B**ehavior Map **Entr**opy, a computationally-efficient approach to Behavior Map entropy maximization based on implicit differentiation. In section 5.3 we show that AMBER monotonically converges to high entropy environments, allowing us to conclude that we contract towards the ground-truth values. In section 6 we provide empirical evidence, showing that AMBER efficiently identifies high-entropy environments and learns R up to high precision, outperforming baselines. All proofs can be found in the appendix.

2. Related work

Inverse Reinforcement Learning Inverse Reinforcement Learning (IRL) is a powerful paradigm for learning Markov Decision Process (MDP) parameters by observing the behavior of a human (Rust, 1994; Ng et al., 2000). A well known problem in IRL is non-identifiability: many different parameters induce the same optimal behavior (Russell, 1998). Thereby, the *true* parameters of the human can not be fully recovered. Previous work in IRL has mostly focused on

[1]Harvard University, Cambridge MA, USA. Correspondence to: Paul Nitschke <pnitschke@g.harvard.edu>.

ICML 2024 Workshop on Models of Human Feedback for AI Alignment.

how to choose a parameter from this set of feasible parameters, rather than resolving the non-identifiability. Common approaches are entropy maximization (Ziebart et al., 2008), maximum margin planning (Ratliff et al., 2006) or adopting a Bayesian perspective (Ramachandran and Amir, 2007). Instead of choosing a parameter from a set of otherwise indistinguishable options, our work narrows in on the true parameters by observing the human in a related scenario, for example under a different transition function.

Active learning of reward functions Actively learning preferences of humans is a well studied problem. Within IRL, previous work has mostly focused on learning a more robust reward function by eliciting the human's feedback under the initial transition dynamics. Lopes et al. (2009); Lindner et al. (2021; 2022) query the human's behavior in different states while Ibarz et al. (2018); Wilde et al. (2021) ask the human to rate different behaviors. The field of Preference Elicitation (PE) (Rashid et al., 2008) studies how to optimally query a human's preferences about different options, such as movie genres. Common approaches are Information Theory (Rokach and Kisilevich, 2012; Canal et al., 2019; Martin et al., 2023) or Bayesian Regret based (Boutilier et al., 2006; Boutilier, 2013). We also aim to ask the human a sequence of informative questions, but we allow the underlying setting to be sequential, unlike in PE.

Environment Design From a theoretic perspective in IRL, it is well known that non-identifiability of the reward function can be resolved up to a constant by observing the human under different transition dynamics or discount rates (Cao et al., 2021). Based on this theoretic insight, Buening et al. (2024) recently proposed Environment Design (ED), assuming an active learning approach to overcome non-identifiability when learning the reward function R. ED proceeds over a sequence of episodes. In each episode, a learner chooses an environment for the human to act in, with the aim of resolving as much uncertainty in R as possible. A large number of environments is generated by randomly changing the transition dynamics from the initial environment, and the chosen environment is the one that maximizes the Bayesian regret of the value function (which is empirically better than other options). The intuition is that we pick an environment where the current reward estimate predicts the behavior of the human as poorly as possible. We instead choose our environment in a computationally-efficient and information-theoretic way, proving that our method chooses a high information-gain environment. We also show our method empirically outperforms Buening et al. (2024).

3. Background

This section provides background on concepts from Inverse Reinforcement Learning and Information Theory, used to derive the intuition and theory behind AMBER.

(Inverse) Reinforcement Learning A Markov Decision Process (MDP) $\mathcal{M}$ is a tuple $(\mathcal{S}, \mathcal{A}, R, T, \gamma)$ where $\mathcal{S}$ is a (finite) state space, $\mathcal{A}$ is a (finite) action space, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a reward function, $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is a transition function and $\gamma \in [0, 1)$ is a discount rate. An optimal human solves for a policy π^* that maximizes the expected, discounted return $J^\pi := \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R_t(s, a)]$: $\pi^* = \max_\pi J^\pi$ (Sutton and Barto, 2018). The value function V^π of a policy π is given as the unique fixed point of the soft Bellman operator $\mathbb{B} : \mathbb{B}V^\pi = \text{logsumexp}_{a \in \mathcal{A}} \big[R(s, a) + \gamma \sum_{s' \in \mathcal{S}} T(s'|s, a)V(s')\big]$. Here, we choose the soft version of the Bellman operator over the hard $\max_{a \in \mathcal{A}}$ version such that we can differentiate through $\mathbb{B}$ (Levine, 2018; Bacon et al., 2019; Nikishin et al., 2022). The Inverse RL problem consists of estimating R given length L observations $\tau_{1:N} := \{(s_0, a_0), (s_1, a_1), ..., (s_{L_i}, a_{L_i})\}_{i=1,...,N}$ of the human (Ng and Russell, 2000).

Definitions and notation We call the unknown parts of R that we aim to learn the human reward parameters $U \subseteq \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}|}$, denote our current belief over U by P_U and the ground-truth human parameters by U_{GT}. We call an MDP without a human parameter an *environment* E, or $\mathcal{M} \backslash U$. Inserting a specific user parameter U into E yields a complete MDP in which we can observe the behavior of the human. We define an intervention $i \in I$ as a perturbation of an environment E to generate a new environment E' and denote the set of all possible interventions by I. For example, we could intervene on an environment by randomly changing the transition function or by doubling all rewards. Finally, we define a behavior as all observations that were generated while pursuing the same goal. Due to stochasticity of the environment, the human may take different paths to reach the same goal, but we do not consider this important. With a slight abuse of notation, we also denote behavior with τ.

A decision rule $d : \tau \to i$ maps behaviors to interventions and the optimal Bayesian decision rule $d^*(l)$ is a decision rule that minimizes a loss function $l : U \times I \to \mathbb{R}$ given a prior P_U over U. The entropy $H(X) := \mathbb{E}[-\log(X)]$ of a random variable X measures the uncertainty of X and the information gain $IG(X|A = a) := H(X) - H(X|A = a)$ measures the reduction in uncertainty of a random variable X by observing another random variable A take on value a. The maximum entropy distribution of a discrete random variable taking $N \in \mathbb{N}$ values has equal probability $\frac{1}{N}$ on all N outcomes.

Behavior maps (BM) A Behavior Map $\mathcal{Y}$ of an environment E maps user parameters $U \sim P_U$ to expected behaviors $\tau_1, ..., \tau_N$ (Ankile et al., 2023).

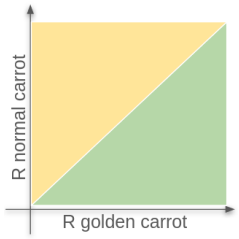

Figure 1. A behavior map

Let $P_{\mathcal{Y}}$ denote the conditional probability of observing behavior τ in environment E given user parameter U:

$$P_{\mathcal{Y}}(E, U) := P[\tau | E, U] \tag{1}$$

Then, the Behavior Map $\mathcal{Y}$ denotes the expected behavior:

$$\mathcal{Y}[U] := \mathbb{E}_{P_{\mathcal{Y}}(E,U)}[\tau], U \tag{2}$$

Figure 1 shows an example Behavior Map for the example in figure 6. For high golden carrot reward values, the rabbit picks the golden carrot (green behavior) while for high normal carrot reward values, the rabbit picks the normal carrot (orange behavior). Note that the distribution in equation 1 is a Dirac distribution for fixed E, U if the human uses a deterministic policy.

4. Problem Setting for Environment Design

In the general Environment Design (ED) problem setting (Buening et al., 2024), we are interested in learning some ground-truth parameters U_{GT} of a human by iteratively observing trajectories over a sequence of related environments. We maintain a belief $P_{U,m}$ over the human parameters, at each episode m, starting with a prior $P_{U,0}$. After observing the human in the initial environment E_0, we update our current belief, yielding $P_{U,1}$.

We know, from literature, that ground-truth parameters are not generally identifiable by observing trajectories in a single environment. Thus, in Environment Design, we observe the human in a chosen new environment and collect additional trajectories to help further identify U_{GT}. We generate a new environment E_{m+1} by slightly perturbing the current one E_m, i.e. with an intervention $i \in I$. For example, an intervention could be to randomly change the transition function of E_m: we need to choose the set of valid interventions. Ideally, we choose a new environment that provides us with the maximum amount of additional information about U_{GT}. We formalize this choice as the following optimization problem:

$$\min_{i \in I} \mathbb{E}_{U \sim P_{U,m}}[l(U, i)] \tag{3}$$

where the loss function $l(U, I) \to \mathbb{R}$ captures our notion of information gained by performing i. For example, the Bayesian regret loss function used in Buening et al. (2024) quantifies how well we can predict the behavior of the human in the new environment E_{m+1}, given our current knowledge $P_{U,m}$. Our method will have a different loss. After observing the human in E_{m+1}, we update our belief to $P_{U,m+1}$. This procedure is repeated for M episodes. We show an overview of the ED framework in appendix 2.

In the above, we make two key assumptions: (i) optimality of the human (otherwise we can't assume that the behavior is caused by the ground-truth human parameters) and (ii) constant ground-truth values across environments (so that we can aggregate information from the same human across different environments).

In summary, there are two main design choices in ED: the loss function l and the set of interventions I. The loss function l must both capture the informativeness of an intervention and be efficiently optimized over I. The set of interventions I must both be able to generate low loss values and generate environments with stationary ground-truth parameters U_{GT}. The remainder of this paper studies our choice of l and I.

5. AMBER: Finding high information gain environments via behavior maps

The Environment Design setup relies on being able to efficiently identify new environments that provide the largest amount of additional information on the unknown parameters. In this section, we characterize these high information gain environments. Using our characterization, we derive a new Environment-Design algorithm, AMBER, for maximizing information gain in each episode.

First, in section 5.1, we connect high information gain environments to Behavior Maps (i.e. the distribution of expected "behaviours" under the current belief) that have high entropy. Second, in section 5.2 we describe how we can maximize the entropy of the Behavior Map. Based on our analysis, we define our algorithm AMBER, which identifies a single high information gain environment for each episode of ED, by maximizing the entropy of corresponding Behavior Maps.

Within each episode, in section 5.3 we show that AMBER monotonically converges to an environment E^* with entropy at least as large as the entropy of a Bernoulli random variable $\mathcal{B}^*$ with $p = 0.5$. That is, AMBER maximizes information gain every episode. Thus, by observing the human in E^*, we reduce the posterior uncertainty by *half* in every iteration. This means that our belief over the ground-truth parameters contracts across multiple episodes when performing ED with environments computed by AMBER.

Finally, we note that while AMBER's theoretical setup is for learning an arbitrary set of parameters U, in this work, we focus only on learning the reward R (instead of including human transition parameters or discount factor, for example).

5.1. High information gain environments have high entropy behavior maps

Suppose we observed the human in an environment E, yielding a posterior P_U. Due to parameter non-identifiability, P_U will likely be spread out. We focus on the region around the posterior mode, which we call the *Region-of-Interest* (ROI). Intuitively, a high-information gain environment E^* is one that cuts down the ROI as much as possible. Equivalently, in a high-information gain environment E^*, each behavior corresponds to a region of the ROI of similar mass. Thus, any behaviour the human chooses will reduce the posterior uncertainty by a maximal amount. Hence, our goal is to find an environment E^* such that the Behavior Map of E^*, when restricted to the ROI, consists of more than two behaviors, with each behavior covering the same percentage of the Behavior Map. Note that this distribution corresponds to the Maximum Entropy Behavior Map. We formalize this intuition in our first theorem (formal statement and proof in Appendix A.1):

Theorem 5.1 (Maximizing the entropy of the behavior map maximizes information gain)**.** *The intervention i^* that maximizes information gain is the intervention that maximizes the entropy of the Behavior map $\mathcal{Y}$, $i^* = \sup_{i \in I} H(\mathcal{Y})$.*

Theorem 5.1 assumes that the policy of the human is deterministic. This is not a strong assumption as there always exists an optimal policy that is deterministic (Sutton and Barto, 2018).

5.2. AMBER: an algorithm for maximizing the entropy of Behavior Maps

Given a posterior P_U and an environment E, we compute the Behavior Map $\mathcal{Y}$ of E restricted to the ROI. Now, assume that the Behavior Map has behaviors $\tau_1, ..., \tau_N$. For behavior τ, we define its 'cover number' of the Behavior Map $\mathbb{C}(\tau|E) \in [0, 1]$, which quantifies the proportion of the Behavior Map that it covers. In Figure 1, the cover number of both behaviors is equal to 0.5 (they cover half the map each). By the previously derived Maximum Entropy principle, our goal is to find an environment E^* such that

$$\mathbb{C}(\tau_1|E^*) = ... = \mathbb{C}(\tau_N|E^*) = \frac{1}{N}. \tag{4}$$

As we assume that the human behaves optimally, $\mathbb{C}(\tau|E)$ corresponds to all $U \sim P_U$ such that the value function is maximal for the respective behavior, where the inequality is meant element-wise:

$$V(\tau_i|E, U) \geq V(\tau_j|E, U), \forall j \in \{1, ..., N\}, i \neq j. \tag{5}$$

Definition 5.2 (Excite and inhibit)**.** Let τ be some behavior and $\mathbb{C}(\tau|E)$ be its cover number. Then, we say an intervention-generated environment E' *excites* the human about τ if $\mathbb{C}(\tau|E')$ increases and *inhibits* the human about τ if $\mathbb{C}(\tau|E')$ decreases.

Now, we increase $H(\mathcal{Y})$ by *exciting* the human about behaviors $\tau_i, i \in \{1, ..., N\}$ which cover less than their maximum entropy share ($\mathbb{C}(\tau_i|E) < \frac{1}{N}$) and *inhibiting* the human about behaviors $\tau_j, j \in \{1, ..., N\}$ that cover more than their respective maximum entropy share ($\mathbb{C}(\tau_j|E) > \frac{1}{N}$). As the cover number $\mathbb{C}(\tau|E)$ of behavior τ corresponds to all user parameters under which the value function restricted to τ is maximal, AMBER increases the value function along states visited by behavior τ_i and decreases the value function along states visited by behavior τ_j by changing R or T:

$$R = \sum_{i=1}^{n} R + \epsilon_1 \mathbb{1}_{\{\mathbb{C}(\tau_i|E) < \frac{1}{N}\}} \nabla_R V\big|_{\tau_i} - \epsilon_1 \mathbb{1}_{\{\mathbb{C}(\tau_i|E) > \frac{1}{N}\}} \nabla_R V\big|_{\tau_i} \tag{6}$$

$$T = \text{softmax}\Big(\sum_{i=1}^{n} T \underbrace{+ \epsilon_1 \mathbb{1}_{\{\mathbb{C}(\tau_i|E) < \frac{1}{N}\}} \nabla_T V\big|_{\tau_i}}_{\text{Excite } \tau_i} \underbrace{- \epsilon_1 \mathbb{1}_{\{\mathbb{C}(\tau_i|E) > \frac{1}{N}\}} \nabla_T V\big|_{\tau_i}}_{\text{Inhibit } \tau_i}\Big) \tag{7}$$

where $\epsilon_1, \epsilon_2 > 0$ are stepsizes and we use a softmax over T such that T remains a probability distribution Nikishin et al. (2022). Note that we focus our experiments on changing T to learn a fixed true R, and so we only consider equation 7. We repeat equation 7 until $\mathbb{C}(\tau_i|E) \approx \frac{1}{N} \forall i = 1, ..., N$ (within a threshold) and provide an overview of AMBER in algorithm 1.

The gradient of V can be computed via implicit differentiation. We note that the AMBER updates above require the computation of ∇V. The gradient of V with respect to R and T can not be directly computed as there exists no closed form expression for V. Instead, we can locally compute it via *Implicit Differentiation* (Bacon et al., 2019; Nikishin et al., 2022).

5.3. Theoretical Properties of AMBER

In the following, we provide theoretical properties of the AMBER algorithm. In particular, we show that AMBER monotonically converges to a high information gain environment within each episode. Thus, across multiple episodes, we can conclude that our posterior contracts towards the true value.

Algorithm 1 Finding a single high information gain environment with AMBER

Input: Current belief P_U, initial environment E, step-sizes $\epsilon_1, \epsilon_2 \in \mathbb{R}^+$
Compute Region of Interest over P_U
while not $\mathbb{C}(\tau_i) \approx \frac{1}{N}\ \forall i = 1, ..., N$ **do**
  Compute Behavior Map $\mathcal{Y}$ of E restricted to Region of Interest, yielding behaviors $\tau_1, ..., \tau_N$
  for $j = 1, ..., N$ **do**
    if $\mathbb{C}(\tau_j) < \frac{1}{N}$ **then**
      $R = R + \epsilon_1 \nabla_R V\big|_{\tau_j}$ {Excite}
      $T = \text{Softmax}\big(T + \epsilon_2 \nabla_T V\big|_{\tau_j}\big)$
    else if $\mathbb{C}(\tau_j) > \frac{1}{N}$ **then**
      $R = R - \epsilon_1 \nabla_R V\big|_{\tau_j}$ {Inhibit}
      $T = \text{Softmax}\big(T - \epsilon_2 \nabla_T V\big|_{\tau_j}\big)$
    end if
  end for
  $E \leftarrow R, T$
end while
Return Maximum Entropy R or T

AMBER balances information gain and generalizability of U. The set of interventions I is in a tradeoff between achieving high information gain (expressive I, or many changes to initial environment) and stationarity of human parameters U_{GT} across different environments (restrictive I, or few changes to initial environment). In the following, we discuss how AMBER's choice of I, i.e. small gradient updates on R or T, balances this conflict.

AMBER generates high information gain environments. First, we argue that if the Behavior Map is not almost surely constant (there are at least two different behaviors), then there always exists a reward function R or transition function T such that the entropy of the Behavior Map $\mathcal{Y}$ is larger or equal to the entropy of a Bernoulli random variable $\mathcal{B}^*$ with $p = 0.5$. We require the Behavior Map to not be constant such that there exists at least one behavior we can excite and a different behavior we can inhibit. The entropy of the Bernoulli random variable $\mathcal{B}^*$ corresponds to *halfing* the Behavior Map:

Theorem 5.3 (Existence of high entropy R or T)**.** *Let $\mathcal{Y}$ be a Behavior Map. If $\mathcal{Y}$ is not almost surely constant, i.e. there exist at least two different behaviors $\tau_i, i = 1, 2, ..., N$, then there exists either a reward function R^* or a transition function T^* such that:*

$$H(\mathcal{Y}|E(R^*)) \geq H(\mathcal{B}^*) \tag{8}$$
$$H(\mathcal{Y}|E(T^*)) \geq H(\mathcal{B}^*), \tag{9}$$

or, there exists an environment such that the resulting Behavior Map is at least halved.

Second, we show that AMBER's learning update generates a sequence of environments such that their information gain is monotonically increasing and converges to an environment with entropy larger or equal to $\mathcal{B}^*$:

Theorem 5.4 (AMBER monotonically converges to high entropy environments)**.** *Let $E^{(n)}, n \in \mathbb{N}$ be a sequence of environments generated via AMBER during the while loop in algorithm 1. Then*

1. *the entropy of their Behavior Maps is monotonically increasing.*
2. *$E^{(n)}$ converges to an environment with Behavior Map entropy larger or equal to $\mathcal{B}^*$.*

AMBER generates environments that are similar to the initial environment. Recall that one of our key assumptions is the stationarity of the human parameters U_{GT} across different environments in ED. In any iteration, if we presented the human with a qualitatively different new environment, then it is no longer reasonable to expect stationarity. Thus, we now verify that AMBER proposes high information gain environments that are similar to the initial environment.

In psychology, the notion of environment, or task, similarity has long been studied, ranging from geometric (Torgerson, 1965) over template matching (Larsen and Bundesen, 1996) to transformational (Imai, 1977) approaches. We follow the work of Tversky (1977), measuring the similarity $\mathbb{S}$ of environments E_1 and E_2 generated by AMBER as a function f of the number of features in common between E_1 and E_2 and the number of features different between E_1 and E_2:

$$\mathbb{S}(E_1, E_2) := f(E_1 \cap E_2) - f(E_1 - E_2) - f(E_2 - E_1) \tag{10}$$

An environment's features are $(\mathcal{S}, \mathcal{A}, R, T)$. By definition, the state and action spaces are equal for environments E_1 and E_2 generated with AMBER and, by lemma A.8 (proportionality of gradients), R and T of the different behaviors are proportional across tasks. Thereby, we have preserved the *gestalt* of the task; this provides support for our assumption that U_{GT} is constant across different environments.

6. Experiments

In our experiments, we aim to answer the following questions: (1) Does AMBER identify high information-gain environments? (2) Does AMBER, paired with the Environment Design paradigm, resolve non-identifiability? (3) Does AMBER converge exponentially fast to the ground-truth parameters U_{GT}? (4) Is AMBER computationally efficient?

6.1. Set-Up

Environments We consider the maze environment from Buening et al. (2024) and aim to learn the values of the

rewards in the top right (R_1) and bottom left (R_2) corner ($U_{GT} := \{R_1, R_2\}$), see the appendix for more details. Hence, we define two behaviors in this environment: choosing R_1 and choosing R_2.

Methods We compare three methods: (i) AMBER, (ii) ED-BIRL from Buening et al. (2024) which randomly inserts walls in the environment and picks the environment with highest Bayesian regret, and (iii) conventional Bayesian IRL BIRL (Ramachandran and Amir, 2007). For comparability, all methods use the log-likelihood of the trajectories and the ROI to calculate the posterior distribution (Herman et al., 2016). Therefore the only differences between the methods are (i) the environments in which we observe the human and (ii) the resulting trajectories of the human. We let each method run for 15 episodes and observe the human once in each episode. All results are averaged across 5 runs.

Metrics We use the following metrics: how often each behavior was chosen (to quantify how informative the environments are), the posterior mean (to quantify accuracy of learning), size of the Region of Interest (to quantify posterior contraction rate) and runtime (to quantify computational efficiency).

6.2. Results

AMBER identifies high information gain environments. Figure 2 shows the cover numbers (left) and resulting entropy (middle) in the maze environment after the first (green), third (blue) and fifth episode (blue). At the start of each episode, the Behavior Map restricted to the ROI is mostly covered by one behavior (behavior 2) which is expected due to non-identifiability. Next, AMBER starts inhibiting the human about behavior 2 until behavior 1 becomes favourable for the first time ($\mathbb{C}(\text{behavior } 1) > 0$). This is repeated until both behaviors cover approximately similar portions of the Behavior Map, indicated by the black, dashed line. Generally, it becomes harder to create equal cover numbers in later episodes as it requires more precise tradeoffs, but this effect is limited as we still reach equal cover numbers quickly. The middle figure in 2 shows the resulting entropy of the Behavior Map during episodes 1, 3 and 5. As the cover numbers of all behaviors are approximately equally large at the end of each episode, the entropy of the Behavior Map monotonically converges to $H(\mathcal{B}^*)$. The right hand side shows how often each behavior is chosen over 15 episodes. In a maximum information gain setting, both behaviors are equally favourable, so both behaviors should be chosen equally often on average. We see that BIRL always chooses the same behavior - as expected as there is no environment design. ED-BIRL does slightly better, sometimes picking the other behavior but can not make both behaviors equally desirable. AMBER, on the other hand, manages to make both behaviors approximately equally attractive, making the human pick both behaviors approximately equally often.

AMBER learns parameters that are non-identifiable in a single environment. The top row in figure 3 shows the mean ratio of $\mathbf{R} := \frac{R_1}{R_2}$ of the posterior distribution over time for BIRL, ED-BIRL and AMBER where $\mathbf{R}$ is evenly spaced between 1 and 3. Here, we report the ratio as it corresponds to the maximum amount of identifiability one can achieve due to reward shaping. In the first example where $\mathbf{R} = 1$, we see that all methods converge to the true value. This is expected as this reward ratio corresponds to the maximum entropy environment where both behaviors are equally desirable. Nonetheless, we note that AMBER converges much faster to the ground-truth value. For all other reward ratios (plots 2-5), BIRL and ED-BIRL converge to roughly the same reward ratio of $\mathbf{R} = 2$. AMBER, on the other hand, converges to the ground-truth value for the ratios 2-4. Only for the largest ratio $\mathbf{R} = 3$ does it not converge to the ground-truth value within 15 episodes. Additionally, note that there is almost no variance in the learning of AMBER as the excite and inhibit part of AMBER is deterministic.

AMBER converges exponentially fast to U_{GT}. The bottom row in figure 3 shows the size of the ROI for different $\mathbf{R}$ values. In our experiments, we set the size of the ROI to 0.8, e.g. the ROI contains 80% of the posterior mass. Thereby, the size of the ROI shrinks in every iteration by at least this factor, indicated by the grey dashed line. The other grey line corresponds to a halfing of the ROI in every iteration, the fastest convergence speed possible if there are only two behaviors. We once again see that there is a difference between $\mathbf{R} = 1$ and $\mathbf{R} \neq 1$. In the $\mathbf{R} = 1$ case (leftmost plot), BIRL and ED-BIRL converge the fastest as they are already in the maximum information gain environment. Interestingly, AMBER converges slightly slower. Nonetheless, no method achieves the fastest possible posterior contraction. For $\mathbf{R} \neq 1$ we see that BIRL and ED-BIRL have the slowest possible convergence rate. This further supports our claim that both methods struggle to find informative environments. AMBER, on the other hand, converges faster than the minimal rate as it observes the human in informative environments. Thereby, we can conclude that AMBER converges exponentially fast to the ground-truth values. Nonetheless, AMBER also doesn't achieve the theoretically optimal contraction rate of $\frac{1}{2}$.

AMBER efficiently finds environments. Table 2 shows the mean wall-clock time in seconds required to generate one environment. BIRL is by far the most efficient as it always observes the human in the base environment. Thereby, the only computation it performs is the calculation of the ROI.

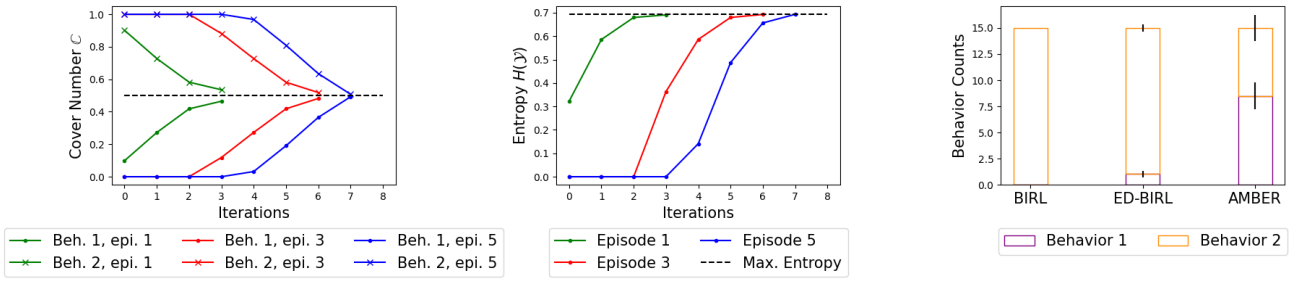

Figure 2. Cover numbers (left figure, equal values are better), resulting entropy (middle figure, higher is better) and behavior counts (right figure, equal values are better) in the maze environment. We see that AMBER takes few iterations to find a good environment within each episode (left two figures), and the human picks the two behaviors in AMBER environments in equal rates (right figure), unlike baselines.

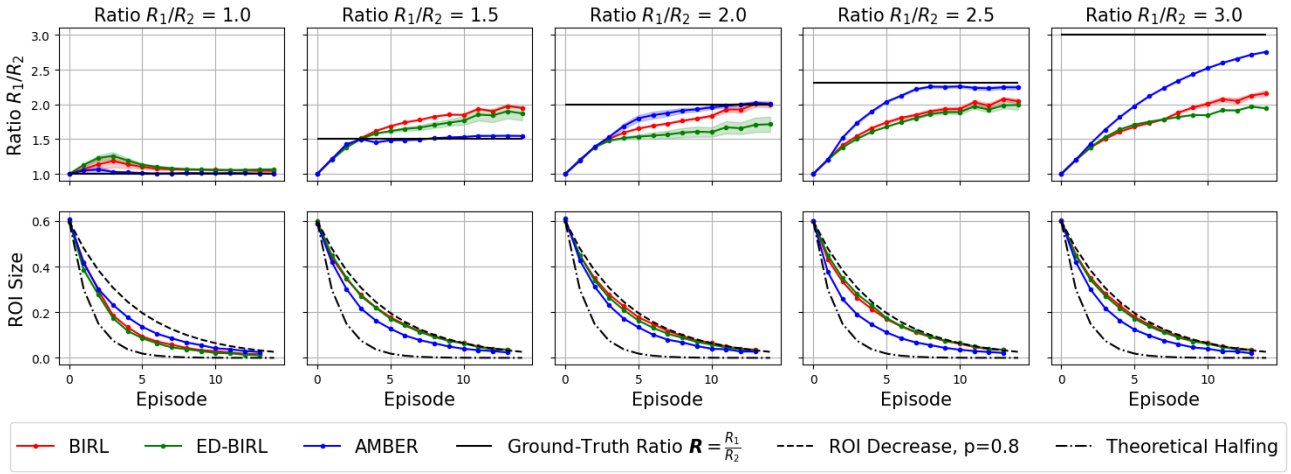

Figure 3. AMBER converges quicker than competing methods to the true rewards. Top row plots the posterior ratio mean of $\mathbf{R} := R_1/R_2$ over episodes, and bottom row plots the size of the ROI over episodes for BIRL, ED-BIRL and AMBER for different ground-truth ratios $\mathbf{R} \in \{1, 1.5, 2, 2.5, 3\}$ (left to right). AMBER is much closer to the theoretical best (halving) than other methods.

Table 1. Mean runtime in seconds to generate one environment

	BIRL	ED-BIRL	AMBER
Runtime	0.27 (0.05)	52.98 (5.78)	5.92 (17.93)

ED-BIRL is computationally more expensive due to the calculation of the Bayesian Regret, where we generated 150 environments in our experiments. AMBER is computationally cheaper because we generate new environments in an informed fashion. Nonetheless, the variance in the run-time is higher as generating high-information gain environments becomes more difficult in later episodes, as we also saw in figure 2.

7. Discussion & Conclusion

In this paper, we introduced a novel Environment-Design algorithm, AMBER, that identifies environments that maximize the information gain for an unknown reward function in an IRL setting. We theoretically and empirically demonstrated that AMBER efficiently identifies high-information gain environments, learns unknown reward functions and outperforms baselines. We conclude the paper by highlighting connections to other domains, as well as limitations and future work.

AMBER is inspired by balanced networks. Our inspiration for AMBER and the terminology we use comes from Balanced Networks from neuroscience (Van Vreeswijk and Sompolinsky, 1996). There, a collection of binary neurons is modeled as two sets of neurons: *excitatory* neurons that fire and pass on information and *inhibitory* neurons that decrease the probability of information being shared. Then, we say that the network is balanced if both populations have the same mean activity. This is akin to AMBER balancing the cover number $\mathbb{C}(\tau|E)$ of the different behaviors.

AMBER identifies low cognitive load environments. AMBER changes the relative value of different behaviors. As the gradient of the value function is proportional to the value function by lemma A.8, AMBER does not change the relative values of states *within* a behavior. Thereby, the human has to not make any inter-behavior decisions again. Rather, the human only has to weigh off entire behaviors against each other. Hence, AMBER identifies environments with low cognitive load (Koppol et al., 2020).

Limitations and future work. The main limitation of AMBER is the computation of the Behavior Map to determine the cover numbers, which we approximated in our experiments with a grid calculation. Each value in the grid corresponds to solving one MDP. Thereby, the number of policy optimizations grows exponentially in the number of unknown parameters. For future work, we are interested in extending our experiments to learning not just rewards R, but also T and γ. From a theoretical perspective, we aim to show that AMBER contracts to U_{GT} and want to better understand to what degree our theorems hold under suboptimality. Finally, from an application perspective, we are interested in how to convert environments generated with AMBER into interpretable, ideally natural language, prompts.

Acknowledgements

The authors thank Ruben Solonch for helpful comments. This material is based upon work supported by the National Science Foundation under Grant No. IIS-2107391. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Lars L Ankile, Brian S Ham, Kevin Mao, Eura Shin, Siddharth Swaroop, Finale Doshi-Velez, and Weiwei Pan. Discovering user types: Mapping user traits by task-specific behaviors in reinforcement learning. *arXiv preprint arXiv:2307.08169*, 2023.

Pierre-Luc Bacon, Florian Schaefer, Clement Gehring, Animashree Anandkumar, and Emma Brunskill. A lagrangian method for inverse problems in reinforcement learning. In *NeurIPS Optimization Foundations for Reinforcement Learning Workshop*, 2019.

Craig Boutilier. Computational decision support: Regret-based models for optimization and preference elicitation, 2013.

Craig Boutilier, Relu Patrascu, Pascal Poupart, and Dale Schuurmans. Constraint-based optimization and utility elicitation using the minimax decision criterion. *Artificial Intelligence*, 170(8-9):686–713, 2006.

Thomas Kleine Buening, Victor Villin, and Christos Dimitrakakis. Environment design for inverse reinforcement learning. In *Proceedings of the 41st International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2024.

Gregory Canal, Andy Massimino, Mark Davenport, and Christopher Rozell. Active embedding search via noisy paired comparisons. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 902–911. PMLR, 09–15 Jun 2019.

Haoyang Cao, Samuel Cohen, and Lukasz Szpruch. Identifiability in inverse reinforcement learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 12362–12373. Curran Associates, Inc., 2021.

Michael Herman, Tobias Gindele, Jörg Wagner, Felix Schmitt, and Wolfram Burgard. Inverse reinforcement learning with simultaneous estimation of rewards and dynamics. In *Artificial intelligence and statistics*, pages 102–110. PMLR, 2016.

Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. Reward learning from human preferences and demonstrations in atari. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Shiro Imai. Pattern similarity and cognitive transformations. *Acta Psychologica*, 41(6):433–447, 1977.

Kuno Kim, Shivam Garg, Kirankumar Shiragur, and Stefano Ermon. Reward identification in inverse reinforcement learning. In *International Conference on Machine Learning*, 2021.

Pallavi Koppol, Henny Admoni, and Reid Simmons. Iterative interactive reward learning. In *Participatory Approaches to Machine Learning, International Conference on Machine Learning Workshop, Virtual*, 2020.

Axel Larsen and Claus Bundesen. A template-matching pandemonium recognizes unconstrained handwritten characters with high accuracy. *Memory & Cognition*, 1996.

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

David Lindner, Matteo Turchetta, Sebastian Tschiatschek, Kamil Ciosek, and Andreas Krause. Information directed reward learning for reinforcement learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 3850–3862. Curran Associates, Inc., 2021.

David Lindner, Andreas Krause, and Giorgia Ramponi. Active exploration for inverse reinforcement learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 5843–5853. Curran Associates, Inc., 2022.

Manuel Lopes, Francisco Melo, and Luis Montesano. Active learning for reward estimation in inverse reinforcement learning. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 31–46. Springer, 2009.

Carlos Martin, Craig Boutilier, and Ofer Meshi. Model-free preference elicitation. In *NeurIPS 2023 Workshop on Adaptive Experimental Design and Active Learning in the Real World*, 2023.

Alberto Maria Metelli, Giorgia Ramponi, Alessandro Concetti, and Marcello Restelli. Provably efficient learning of transferable rewards. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7665–7676. PMLR, 18–24 Jul 2021.

Alberto Maria Metelli, Filippo Lazzati, and Marcello Restelli. Towards theoretical understanding of inverse reinforcement learning. In *International Conference on Machine Learning*, pages 24555–24591. PMLR, 2023.

Andrew Ng and Stuart Russell. Algorithms for inverse reinforcement learning. *ICML '00 Proceedings of the Seventeenth International Conference on Machine Learning*, 05 2000.

Andrew Y Ng, Stuart Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2, 2000.

Evgenii Nikishin, Romina Abachi, Rishabh Agarwal, and Pierre-Luc Bacon. Control-oriented model-based reinforcement learning with implicit differentiation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7886–7894, 2022.

Deepak Ramachandran and Eyal Amir. Bayesian inverse reinforcement learning. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, page 2586–2591, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

Al Mamunur Rashid, George Karypis, and John Riedl. Learning preferences of new users in recommender systems: an information theoretic approach. *Acm Sigkdd Explorations Newsletter*, 10(2):90–100, 2008.

Nathan D. Ratliff, J. Andrew Bagnell, and Martin Zinkevich. Maximum margin planning. In *Proceedings of the 23rd International Conference on Machine Learning*, volume 148 of *ACM International Conference Proceeding Series*, pages 729–736. ACM, 2006. ISBN 1-59593-383-2. doi: 10.1145/1143844.1143936.

Lior Rokach and Slava Kisilevich. Initial profile generation in recommender systems using pairwise comparison. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):1854–1859, 2012.

Stuart Russell. Learning agents for uncertain environments (extended abstract). In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, page 101–103, New York, NY, USA, 1998. Association for Computing Machinery. ISBN 1581130570. doi: 10.1145/279943.279964.

John Rust. Chapter 51 structural estimation of markov decision processes. volume 4 of *Handbook of Econometrics*, pages 3081–3143. 1994. doi: https://doi.org/10.1016/S1573-4412(05)80020-0.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

Warren S. Torgerson. Multidimensional scaling of similarity. *Psychometrika*, 1965.

Amos Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, 1977. doi: 10.1037/0033-295X.84.4.327.

Carl Van Vreeswijk and Haim Sompolinsky. Chaos in neuronal networks with balanced excitatory and inhibitory activity. *Science*, 274(5293):1724–1726, 1996.

Nils Wilde, Erdem Bıyık, Dorsa Sadigh, and Stephen L Smith. Learning reward functions from scale feedback. *arXiv preprint arXiv:2110.00284*, 2021.

Chao Yu, Jiming Liu, and Hongyi Zhao. Inverse reinforcement learning for intelligent mechanical ventilation and sedative dosing in intensive care units. *BMC Medical Informatics and Decision Making*, 19, 04 2019. doi: 10.1186/s12911-019-0763-6.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

A. Appendix

The appendix is structured as follows:

A.1. Proof: maximum information gain environments have maximum entropy behavior maps

Theorem A.1 (Maximizing the Entropy of the Behavior Map Maximizes Information Gain)**.** *Suppose the policy of the human is deterministic and we use the following loss function:*

$$l(u, i) := -IG(P_U|\pi^*[i, U_{GT}]) \tag{11}$$

e.g. the loss of observing the human in a sub-optimal environment is having low information gain. Here, $\pi^*[i, U_{GT}]$ is a random variable that returns the behavior of the human given an environment generated via intervention i and their ground-truth parameters U_{GT}.
Then, we have

$$i^* := \arg\min_{i \in I} d^*(l) = \sup_{i \in i} H(\mathcal{Y}) \tag{12}$$

The optimal intervention is the intervention that maximizes the entropy of the Behavior Map.

Proof.

$$\begin{aligned}
i^* &= \arg\min_{i \in I} \mathbb{E}_{u \sim P_U}[-IG(P_U|\pi^*[i, u])] && (13)\\
&= \arg\max_{i \in I} \mathbb{E}_{u \sim P_U}[H(P_U) - H(P_U|\pi^*[i, u] = \tau)] && (14)\\
&= \arg\min_{i \in I} \mathbb{E}_{u \sim P_U}[H(P_U|\pi^*[i, u] = \tau)] && (15)\\
&= \arg\max_{i \in I} \mathbb{E}_{u \sim P_U}[\underbrace{H(\pi^*[i, u] = \tau|P_U)}_{=0 \text{ as } \tau \text{ is constant}} - H(\pi^*[i, u]) + \underbrace{H(P_U)}_{\text{Independent of } i}], \text{ Bayes Rule for Cond. Entropy.} && (16)\\
&= \arg\max_{i \in I} \mathbb{E}_{u \sim P_U}[H(\pi^*[i, u])] && (17)\\
&= \arg\max_{i \in I} \mathbb{E}_{u \sim P_U}[\mathbb{E}_{\tau \sim \pi^*[i,u]}[-\log(\tau)]] && (18)\\
& && (19)
\end{aligned}$$

Next up, for a fixed $i \in I$, note that the random variable $\pi^*[i, U_{GT}]$ is almost surely constant as the policy of the human is deterministic. Using the convention $0 \cdot \log(0) := 0$ thereby yields:

$$\begin{aligned}
\arg\max_{i \in I} \mathbb{E}_{u \sim P_U}[\mathbb{E}_{\tau \sim \pi^*[i,u]}[-\log(\tau)]] &= \arg\max_{i \in I} \mathbb{E}_{u \sim P_U}[-\log(\mathbb{E}_{\tau \sim \pi^*[i,u]}[\tau])] && (20)\\
&= \arg\max_{i \in I} \mathbb{E}_{u \sim P_U}[-\log(\mathcal{Y})] && (21)\\
&= \arg\max_{i \in I} H(\mathcal{Y}) && (22)
\end{aligned}$$

□

A.2. Proof: AMBER converges to high entropy behavior maps

This section aims to prove that AMBER monotonically converges to high entropy environments. To this end, we first define the cover number $\mathbb{C}(\tau)$ of a behavior τ in definition A.2. $\mathbb{C}(\tau)$ measures what proportion of the Behavior Map is covered by the behavior τ and quantifies the entropy of the Behavior Map. Second, we define in definition A.3 the AMBER learning update which increases entropy by exciting and inhibiting behavior. Third, we show that the cover number has useful properties when using the AMBER update, namely it is monotonic, continuous and converges to 1 (excite) or 0 (inhibit), section A.4. Fourth, using these properties we can prove that there exist high entropy reward and transition functions R^* and T^*, theorem A.5, and that AMBER monotonically converges to R^* and T^*, theorem A.7.
To keep notation concise, we prove all theorems for the reward function R. However, all results also hold with identical arguments for T.

Definition A.2 (Cover number)**.** Given a Behavior Map Random Variable $\mathcal{Y}$ for a fixed reward function R, define for behavior $\tau_i, i = 1, ..., N$ its cover number $\mathbb{C}(R, \tau_i)$:

$$\mathbb{C}(R, \tau_i) := \frac{\int_u \mathbb{1}_{\{V(\tau_i|R,u) > V(\tau_j|R,u) \forall j=1,...,N, i \neq j\}} du}{\int_u 1 du} \tag{23}$$

here $N \in \mathbb{N}$ is the number of different behaviors which is finite as the state and action spaces are finite and the inequality is meant element-wise.

Definition A.3 (AMBER update)**.** Define AMBER 's update procedure:

$$\text{Excite: } f_+(R, \tau) := R + \epsilon \nabla_R V|_\tau \tag{24}$$
$$\text{Inhibit: } f_-(R, \tau) := R - \epsilon \nabla_R V|_\tau \tag{25}$$

where R is a reward function, τ is behavior, $\epsilon > 0$ is a step size and V is a value function under a greedy policy given an arbitrary reward and transition function.

Theorem A.4 (Properties of the cover number)**.** *If R is such that $\nabla_R V \geq c > 0$ for some universal constant c, then the cover number $\mathbb{C}(R, \tau)$ has the following properties :*

1. *(i)* ***Probability:*** $\mathbb{C}(R, \tau) \in [0, 1]$
2. *(ii)* ***Law of Total Probability:*** $\sum_{i=1}^{N} \mathbb{C}(R, \tau) = 1$
3. *(iii)* ***Monotonicity:*** $\mathbb{C}(f_+(R), \tau) \geq \mathbb{C}(\tau)$ *and* $\mathbb{C}(f_-(R), \tau_i) \leq \mathbb{C}(\tau)$
4. *(iv)* ***Continuity:*** $\mathbb{C}(R, \tau)$ *is continuous in* R
5. *(v)* ***Asymptotic Behavior:*** $\lim_{n\to\infty} \mathbb{C}(f_+^n(R), \tau) = 1$ *and* $\lim_{n\to\infty} \mathbb{C}(f_-^n(R), \tau) = 0$

Proof. **(i) Probability:** Definition.
(ii) Law of Total Probability: Linearity.
(iii) Monotonicity:
First, note that

$$V(s|f_+(R), u) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k f_+(R)_k | s, u\right], \text{ Definition} \tag{26}$$
$$= \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R_k + (\epsilon \nabla_R V|_{\tau_i})_k | s, u\right], \text{ Definition} \tag{27}$$
$$= \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R_k | s, u\right] + \epsilon \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k (\nabla_R V|_{\tau_i})_k | s, u\right], \text{ Linearity} \tag{28}$$
$$= V(s|R, u) + \epsilon V(s|\nabla_R V|_{\tau_i} u), \text{ Definition} \tag{29}$$
$$\geq V(s|R, u) + \epsilon V(s|c, u), \text{ Assumption A.4} \tag{30}$$
$$= V(s|R, u) + \frac{\epsilon}{1-\lambda} c \mathbb{1}_{\{s \in \tau\}} \tag{31}$$
$$\geq V(s|R, u) \tag{32}$$

Thereby, for the set $\bar{U}(\tau_i, R) := \{u \in U : V(\tau_i|R, u) \geq V(\tau_j|R, u) \forall j = 1, ..., N, i \neq j\}$ we have that

$$\bar{U}|(\tau_i, R) \subseteq \bar{U}|(\tau_i, f^+(R)) \tag{33}$$

so the claim follows by the definition of the cover number. Here, the set $\bar{U}(\tau_i, R)$ corresponds to all parameters $u \in U$ such that behavior τ is optimal under that parameter. We can do an identical argument for f_-, yielding the second claim.

(iv) Continuity:
Let R_n be a sequence of reward functions such that $R_n \to R, n \to \infty$. To show (iv), we show that

$$\lim_{n\to\infty} \mathbb{C}(R_n, \tau_i) = \mathbb{C}(R, \tau_i) \tag{34}$$

In the following, we denote by DCT the Dominated Convergence Theorem. W.l.o.g. we can assume that $\exists N' \in \mathbb{N}, c' \in \mathbb{R}$: $|R_n| \leq c' \in \mathbb{R}\ \forall n \geq N'$ where the inequality holds element-wise. Now, we have that

$$\lim_{n\to\infty} \mathbb{C}(R_n, \tau_i) = \lim_{n\to\infty, n\geq N'} \mathbb{C}(R_n, \tau_i) \tag{35}$$

$$= \lim_{n\to\infty, n\geq N'} \frac{\int_u \mathbb{1}_{\{V(\tau_i|R_n,u)>V(\tau_j|R_n,u)\forall j=1,\dots,N, i\neq j\}} du}{\int_u 1 du} \text{, Definition} \tag{36}$$

$$= \frac{\int_u \lim_{n\to\infty, n\geq N'} \mathbb{1}_{\{V(\tau_i|R_n,u)>V(\tau_j|R_n,u)\forall j=1,\dots,N, i\neq j\}} du}{\int_u 1 du} \text{, DCT} \tag{37}$$

$$= \frac{\int_u \mathbb{1}_{\lim_{n\to\infty, n\geq N'}\{V(\tau_i|R_n,u)>V(\tau_j|R_n,u)\forall j=1,\dots,N, i\neq j\}} du}{\int_u 1 du} \text{, Set Theoretic Limit} \tag{38}$$

$$= \frac{\int_u \mathbb{1}_{\lim_{n\to\infty, n\geq N'}\{V(\tau_i|R_n,u)\geq V(\tau_j|R_n,u)\forall j=1,\dots,N, i\neq j\}} du}{\int_u 1 du} \text{, } \tau_j \neq \tau_i. \tag{39}$$

$$= \frac{\int_u \mathbb{1}_{\{\lim_{n\to\infty, n\geq N'} V(\tau_i|R_n,u)\geq \lim_{n\to\infty, n\geq N'} V(\tau_j|R_n,u)\forall j=1,\dots,N, i\neq j\}} du}{\int_u 1 du} \text{, Sandwich} \tag{40}$$

$$= \frac{\int_u \mathbb{1}_{\{\lim_{n\to\infty, n\geq N'} \mathbb{E}[\sum_{k=0}^{\infty} \gamma^k R_k^n|\tau_i,u]\geq \lim_{n\to\infty, n\geq N'} \mathbb{E}[\sum_{k=0}^{\infty} \gamma^k R_k^n|\tau_j,u]\forall j=1,\dots,N, i\neq j\}} du}{\int_u 1 du} \tag{41}$$

$$= \frac{\int_u \mathbb{1}_{\{\mathbb{E}[\lim_{n\to\infty, n\geq N'} \sum_{k=0}^{\infty} \gamma^k R_k^n|\tau_i,u]\geq \mathbb{E}[\lim_{n\to\infty, n\geq N'} \sum_{k=0}^{\infty} \gamma^k R_k^n|\tau_j,u]\forall j=1,\dots,N, i\neq j\}} du}{\int_u 1 du} \text{, DCT} \tag{42}$$

$$= \frac{\int_u \mathbb{1}_{\{\mathbb{E}[\sum_{k=0}^{\infty} \lim_{n\to\infty, n\geq N'} \gamma^k R_k^n|\tau_i,u]\geq \mathbb{E}[\sum_{k=0}^{\infty} \lim_{n\to\infty, n\geq N'} \gamma^k R_k^n|\tau_j,u]\forall j=1,\dots,N, i\neq j\}} du}{\int_u 1 du} \tag{43}$$

$$= \frac{\int_u \mathbb{1}_{\{\mathbb{E}\sum_{k=0}^{\infty} \gamma^k R_k|\tau_i,u]\geq \mathbb{E}[\sum_{k=0}^{\infty} \gamma^k R_k|\tau_j,u]\forall j=1,\dots,N, i\neq j\}} du}{\int_u 1 du} \text{, } R_n \to R \tag{44}$$

$$= \frac{\int_u \mathbb{1}_{\{\mathbb{E}\sum_{k=0}^{\infty} \gamma^k R_k|\tau_i,u]> \mathbb{E}[\sum_{k=0}^{\infty} \gamma^k R_k|\tau_j,u]\forall j=1,\dots,N, i\neq j\}} du}{\int_u 1 du} \text{, } \tau_j \neq \tau_i. \tag{45}$$

$$= \frac{\int_u \mathbb{1}_{\{V(\tau_i|R,u)>V(\tau_j|R,u)\forall j=1,\dots,N, i\neq j\}} du}{\int_u 1 du} \tag{46}$$

$$= \mathbb{C}(R, \tau_i) \text{ , Definition} \tag{47}$$

(v) Asymptotic behavior:
With an argument identical to (iii), we have that

$$V(s|f_+^n(R), u) \geq V(s|R, u) + \frac{n}{1-\lambda} c \mathbb{1}_{\{s\in\tau\}} \tag{48}$$

Thereby, we have that $V(s|f_+^n(R))|_\tau = \infty$ for $n \to \infty$ and thereby

$$\bar{U}|(\tau_i, V(s|f_+^\infty(R)) = U \tag{49}$$

which yields the desired claim. The argument for f^- works the same way. □

With these auxiliary results done, we can come to the second main result:

Theorem A.5 (Existence of high entropy environments)**.** *Let $\mathcal{Y}$ be a Behavior Map random variable. If $\mathcal{Y}$ is not almost surely constant, e.g. there exist at least two different behaviors $\tau_i, i \in \{1, 2, ...\}$, then there exists a reward function R^* or a transition function T^* such that*

$$H(\mathcal{Y}|R^*) \geq H(\mathcal{B}^*) \tag{50}$$

where $\mathcal{B}^*$ is a Bernoulli random variable with $p = 0.5$, e.g. there exists a reward function R^* such that the resulting Behavior Map is at least halved.

Proof. First, assume that there exist exactly two behaviors τ_1, τ_2. W.l.o.g. assume that $\mathbb{C}(R, \tau_1) < 0.5$. Now, by property (v), $\lim_{n\to\infty} \mathbb{C}(f_+^n(R), \tau_1) = 1$. By continuity (iii), the Mean Value Theorem yields that there exists an R^* such that $\mathbb{C}(R^*, \tau_1) = 0.5$. Here, the Mean Value Theorem holds as the reward function is finite dimensional, as $\mathcal{S}, \mathcal{A}$ are finite by assumption. Now, property (ii) yields that $\mathbb{C}(R^*, \tau_2) = 0.5$. Thereby, $P(\mathcal{Y} = \tau_1) = P(\mathcal{Y} = \tau_2) = 0.5$ and hence $H(\mathcal{Y}|R^*) = H(B)$.

Next, assume that there exist more than two behaviors:

$$\mathcal{Y} \in \{\tau_1, ..., \tau_N\}, N \geq 3, \text{ almost surely} \tag{51}$$

Now, define a *reduced* Behavior Map $\tilde{\mathcal{Y}}$:

$$\tilde{\mathcal{Y}} = \tau_1 \mathbb{1}_{\{\mathcal{Y}=\tau_1\}} + \tau_2 \mathbb{1}_{\{\mathcal{Y}\neq\tau_1\}} \tag{52}$$

Then, we have that $H(\tilde{\mathcal{Y}}) \leq H(\mathcal{Y})$ and by the previous argument $H(\tilde{\mathcal{Y}}) = H(\mathcal{B}^*)$ so

$$H(\mathcal{Y}) \geq H(\mathcal{B}^*) \tag{53}$$

□

Now, the previous theorem only yields that there exists a reward function R^* that halves the BM. Next, we want to show that AMBER *converges* to R^*.
To this end, define the AMBER learning procedure h:

Definition A.6 (AMBER Learning)**.** Let $\mathcal{Y}$ be a Behavior Map random variable and $\tau_1, ..., \tau_N, N \in \mathbb{N}$ be the observed behavior. Then, define our learning procedure:

$$h(R) := \sum_{i=1}^{N} f^-(R, \tau_i) \mathbb{1}_{\{\mathbb{C}(R,\tau_i) > \frac{1}{N}\}} \tag{54}$$

, e.g. we *inhibit* all behaviors that cover more that their maximum entropy share of the Behavior Map.

In definition A.6, we only inhibit behavior and don't excite behavior as it simplifies the notation. In practice, we both excite and inhibit to converge faster.

Theorem A.7 (AMBER monotonically converges to high entropy environments.)**.** *Let $E^{(0)}$ be an environment and $E^{(n)}, n \in \mathbb{N}$ be a sequence of environments generated via A.6. If all environments allow for at least two behaviors, then we have:*

$$H(\mathcal{Y}|E^{(n+1)}) \geq H(\mathcal{Y}|E^{(n)}) \tag{55}$$

$$\lim_{n\to\infty} H(\mathcal{Y}|E^{(n)}) \geq H(\mathcal{B}^*) \tag{56}$$

Proof. We first show that the entropies of the Behavior Maps are monotonically increasing.
To this end, let $E^{(n)}$ be an environment generated via A.6 with N different behaviors and $\tau_1, ..., \tau_M$ be its behaviors with cover numbers larger than $\frac{1}{N}$. Applying the AMBER update once decreases or keeps the cover number of all behaviors $\tau_1, ..., \tau_M$ equal by (iii) in theorem A.4. Due to (ii) in theorem A.4, the cover numbers of all other behaviors must thereby increase or stay equal from which the claim follows.

To show the second statement, we first note that the entropy is upper bounded as there are only finitely many optimal behaviors. As the sequence $H(\mathcal{Y}|E^{(n)})$ is monotonic by the previous argument, it thereby converges to some environment E' with entropy H'. Now suppose, for sake of contradiction, that $H^{'} < H(\mathcal{B}^*)$. Let $\tau_1, ..., \tau_N, N \geq 2$ be the behaviors of E'. As $H^{'} < H(\mathcal{B}^*)$, there must exist a behavior τ^* with cover number $\mathbb{C}(\tau^*) > \frac{1}{N}$. With an argument identical to the previous one, performing another AMBER update yields an environment $E^{''}$ where the cover number of τ^* decreased. Thereby, we have that $H^{''} > H^{'}$, a contradiction. □

A.3. Proof: proportional gradients

Lemma A.8 (Proportionality of gradients)**.** *Let V, R, T be value, reward and transition function, respectively. Let π be a Boltzmann policy. Then the value function is approximately proportional to its gradient w.r.t. the reward and transition function:*

$$V^\pi \propto \nabla_R V^\pi, \quad V^\pi \propto \nabla_T V^\pi \tag{57}$$

Proof. Recall that the Bellman equation for the value function is given by:

$$V^\pi(s) = \sum_{a \in A} \pi(a|s) \sum_{s' \in S} T(s'|s,a) \left[R(s,a) + \gamma V^\pi(s')\right] \tag{58}$$

Solving it for the value function yields:

$$V^\pi = (I - \gamma P^\pi)^{-1} R^\pi \tag{59}$$

where T^π is the state transition matrix under policy π, with elements $P^\pi(s,s') = \sum_{a \in A} \pi(a|s) T(s'|s,a)$ and R^π is the expected reward vector under policy π, with elements $R^\pi(s) = \sum_{a \in A} \pi(a|s) R(s,a)$. Thereby, we have:

$$\frac{\partial V^\pi}{\partial R^\pi} = (I - \gamma P^\pi)^{-1} \tag{60}$$

Since $R^\pi(s) = \sum_{a \in A} \pi(a|s) R(s,a)$, taking the derivative with respect to $R(s,a)$ yields:

$$\frac{\partial R^\pi(s)}{\partial R(s,a)} = \pi(a|s) \tag{61}$$

Combining the two derivatives using the chain rule, we have:

$$\frac{\partial V^\pi}{\partial R(s,a)} = \frac{\partial V^\pi}{\partial R^\pi} \cdot \frac{\partial R^\pi}{\partial R(s,a)} = (I - \gamma P^\pi)^{-1} \cdot \pi(a|s) \tag{62}$$

As the human uses a Boltzmann policy, we have:

$$\frac{\partial V^\pi}{\partial R(s,a)} = \frac{\partial V^\pi}{\partial R^\pi} \cdot \frac{\partial R^\pi}{\partial R(s,a)} = (I - \gamma P^\pi)^{-1} \cdot e^{\beta Q^*(s,a)} \tag{63}$$

Thereby, the result follows as the Value-Function is approximately proportional to the Q-Function.
We can use an identical argument for the transition function to yield:

$$\begin{aligned} \frac{\partial V^\pi(s)}{\partial P(s'|s,a)} &= \gamma (I - \gamma P^\pi)^{-1} R^\pi (I - \gamma P^\pi)^{-1} \pi(a|s) && (64) \\ &= \gamma V^\pi(s) (I - \gamma P^\pi)^{-1} \pi(a|s) && (65) \\ &\propto V^\pi(s) && (66) \end{aligned}$$

□

Figure 4 shows an example of theorem A.3.

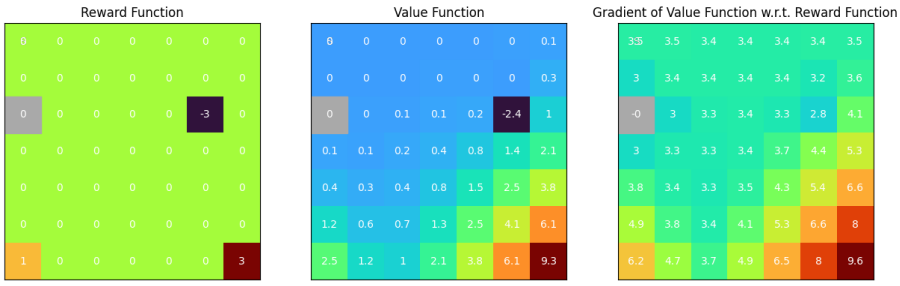

Figure 4. Grid world with reward function (left), value function (middle) and gradient of the value function with respect to the reward function (right)

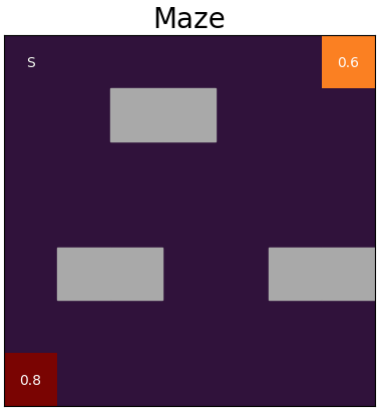

Figure 5. Maze Environments used. We aim to learn the values of the rewards in the top right and bottom corner.

A.4. Further details on the experiments

This section provides further details on the experiments.

Maze environment We consider the maze environment from (Buening et al., 2024) and aim to learn the value of the reward function in the top right and bottom left corner, figure 5. The human starts in the top left corner. They transition into the intended state with probability $p = 0.9$ and "slip" into an adjacent state with probability $p = 0.1$. Their discount rate is $\gamma = 0.8$. In each episode, we observe the human once.

Hyperparameters Table 2 shows the hyperparameters used in the empirical section.

A.5. Environment Design pseudocode

Algorithm 2 shows the general Environment Design algorithm.

Table 2. Hyperparameters for experiments

Method	Parameter	Value
AMBER	Stepsizes ϵ_1, ϵ_2	0.0001
	Behavior Map mesh size	20
ED-BIRL	# environments	150
	# walls	5
	Monte Carlo sampler	Metropolis-Hastings (default)
	# Monte Carlo samples	400
	# Monte Carlo burn-In	150

Algorithm 2 Environment Design Framework

Input: Prior P_U, initial environment E
Input: Interventions I, loss function $l(U, I)$
$E_1, P_{U,1} \leftarrow E, P_U$
for $m = 1, ..., M$ **do**
  Observe human in E_m to get τ_m
  Update belief $P_{U,m}|(\tau_1, ..., \tau_m)$
  Find optimal intervention: $i^* = \min_{i \in I} \mathbb{E}_{U \sim P_{U,m}}[l(U, i)]$
  Design new environment E_{m+1} with i^*
end for
Return posterior $P_{U,M}$

B. Additional Figures

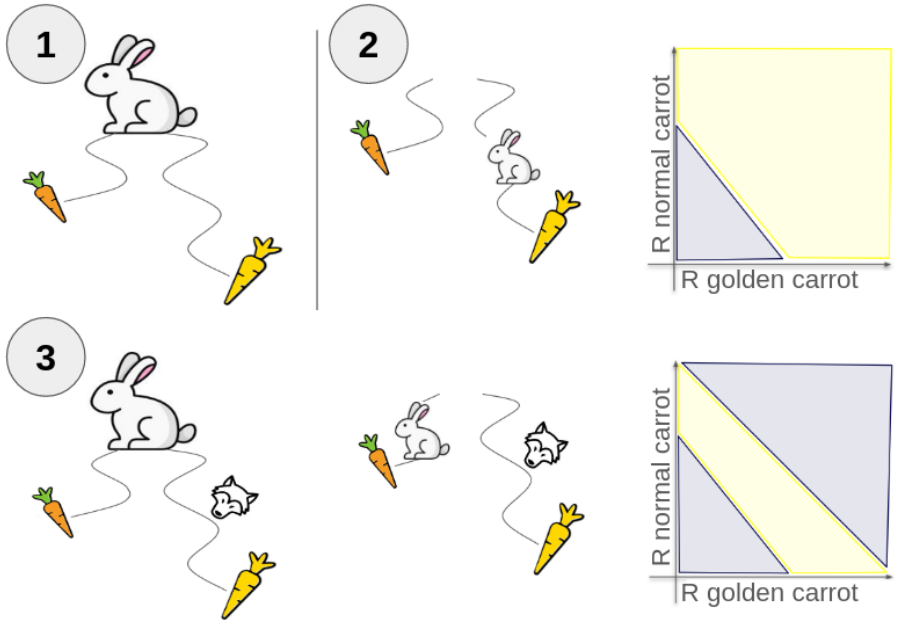

Figure 6. Visual overview of AMBER (Panel 1). A rabbit chooses between a nearby carrot and a distant golden carrot. We want to learn how the rabbit values the carrots. Observing the rabbit pick the more favourable golden carrot (Panel 2) only slightly reduces our uncertainty about R. In the density plots (right hand side), yellow corresponds to high probability while blue is low probability. Based on the uncertainty in the yellow region, AMBER (Panel 3) generates a related scenario by perturbing the transition that makes both carrots equally desirable: A wolf was seen near the golden carrot. Thereby, the rabbit has to make a harder, and hence more informative, choice. Observing the rabbit pick the normal carrot over the golden carrot thereby halves the uncertainty about R. Iterating this procedure yields exponentially fast convergence to the ground-truth reward parameters.