Environment Design for Biased Decision Makers

Guanghui Yu and **Chien-Ju Ho**
Washington University in St. Louis
{guanghuiyu, chienju.ho}@wustl.edu

Abstract

We study the *environment design* problem for biased decision makers. In an environment design problem, an informed *principal* aims to update the decision making environment to influence the decisions made by the *agent*. This problem is ubiquitous in various domains, e.g., a social networking platform might want to update its website to encourage more user engagement. In this work, we focus on the scenario in which the agent might exhibit biases in decision making. We relax the common assumption that the agent is rational and aim to incorporate models of biased agents in environment design. We formulate the environment design problem under the Markov decision process (MDP) and incorporate common models of biased agents through introducing general time-discounting functions. We then formalize the environment design problem as constrained optimization problems and propose corresponding algorithms. We conduct both simulations and real human-subject experiments with workers recruited from Amazon Mechanical Turk to evaluate our proposed algorithms.

1 Introduction

We explore the problem where two parties with mis-aligned objectives, a *principal* and an *agent*, are in the same sequential decision making environment. The goal of the agent is to take a sequence of actions to maximize his total payoff[1]. The principal cannot directly take actions but can update the environment to influence the agent's actions and receive reward based on the agent's actions. The goal of the principal is to update the environment such that the agent takes actions that maximize the principal's payoff.

This problem setting is motivated by several existing and potential applications. For example, a user-generated content website might want to update their site to provide incentives, such as badges or virtual points, to encourage users to consume and rate the content on their website. An online retailer might want to decide when and whether to provide coupons to nudge the user to make the purchase. An assistive AI agent might want to provide interventions, such as reminding messages, to help humans achieve personal goals, such as reducing the amount of time spent on social networking sites.

If we assume the agent is *rational* and makes decisions according to the optimal policy, this problem is similar to several existing works in the literature, including policy teaching [Zhang and Parkes, 2008; Zhang *et al.*, 2018], in which the principal updates the reward functions to induce the agent to take certain policies, and the poisoning attack for reinforcement learning [Rakhsha *et al.*, 2020; Zhang *et al.*, 2020], in which an adversarial principal aims to modify the training environment such that the agent learns the undesired policy. In this work, we are motivated by the natural setting in which the agent is a human being and might exhibit biases in decision making. As observed in empirical studies, humans are known to exhibit systematic biases in making decisions. For example, humans might not have the ability to reason far ahead into the future [Kahneman, 2003] or might exhibit *present bias* [O'Donoghue and Rabin, 1999], giving stronger weights on immediate costs and benefits rather than balancing them against those in the future.

We study this two-party sequential decision making problem under the formulation of Markov decision process (MDP). A standard MDP is characterized by the set of states, the set of actions, the state transition function, and the reward function. The solution of an MDP is a *policy* that specifies which action to take in each state that maximizes the total reward. Our setting deviates from the standard MDP in two perspectives. First, there are two parties, a principal and an agent, in the same decision making environment. The principal and the agent share the same information about the state, state transition, and action set. However, they have different reward functions. Moreover, while the agent can take actions in the environment, the principal can only update the environment to influence the agent's actions. Second, the agent exhibits decision-making biases in his solution to the MDP. Since the focus of this paper is in sequential decision making, we focus on the *time-related decision biases*, including myopic decision making, bounded rationality, and present bias.

We consider two natural sets of design spaces that the principal can choose from to update the environment. In the first design space, the principal can modify the agent's reward function in MDP, and the agent's policy is based on the modified reward function. This design space corresponds to the

[1] We use *she* to denote the principal and *he* to denote the agent.

scenario in which the principal can update the environment in a global manner (e.g., changing the badge design in social networking sites), and the agent will take actions in the updated environment. In the second design space, when the agent is choosing an action during decision time, the principal can offer additional incentives to nudge the agent to choose a different action. This design space corresponds to the scenario in which the principal can take interventions during the agent's decision time (e.g., offering a coupon when the user navigates to a certain page). In environment design with both design spaces, the goal of the principal is to maximize her own total rewards, depending on the principal's reward function and the agent's actions, subject to a budget constraint that the amount of environment updates is limited.

We formulate the principal's environment design problems as constrained optimization problems under both design spaces. We first show that the optimization problems are generally NP-hard to solve for both design spaces. We then propose relaxed formulations and corresponding algorithms for solving the problems. To evaluate the effectiveness of our proposed algorithms for environment design, we conduct simulations to understand the algorithm performance over a range of scenarios and parameters. Moreover, to examine whether we can indeed update the environment to influence the decisions of real-world human decision makers, we conduct a human-subject experiment with 300 workers from Amazon Mechanical Turk. Our results demonstrate the environment updates derived by our algorithms can effectively influence humans' decisions and lead to better total payoff.

Contributions. The main contributions of this work can be summarized as follows.

- We formulate the environment design problem with biased decision makers. To the best of our knowledge, this is the first work to incorporate human behavioral models in environment design, i.e., modifying the MDP environment to change the decisions made by humans with decision biases.
- We show that the environment design problems are generally NP-hard to solve. We propose algorithms for two different sets of design spaces, reward function modification and action nudge.
- We have conducted simulations and real-world human-subject experiments with 300 human subjects to evaluate our proposed algorithms for environment design. The results demonstrate that our algorithms lead to better outcomes with biased decision makers.

1.1 Related Work

Our work is built on the formulation of Markov decision process (MDP) commonly seen in reinforcement learning. Instead of solving the agent's optimal policy, we consider a Stackelberg game formulation, in which the principal first chooses how to update the environment, and then the agent makes decisions in the updated environment. The closest works that consider this two-party setting in MDP include policy teaching [Zhang and Parkes, 2008; Zhang *et al.*, 2009, 2018] and poisoning attack for reinforcement learning [Rakhsha *et al.*, 2020; Zhang *et al.*, 2020]. Our work deviates from these works by incorporating human behavioral models in the framework. The human models considered in this work are empirically motivated from behavioral economics, such as *bounded rationality* [Kahneman, 2003] and *present bias* [O'Donoghue and Rabin, 1999].

Our work joins the recent research theme that incorporates human models in computational frameworks [Frazier *et al.*, 2014; Mansour *et al.*, 2015; Tang and Ho, 2019, 2021; Kleinberg and Oren, 2014; Masters *et al.*, 2021a,b]. There have been other lines of research that also includes humans in the loop of reinforcement learning frameworks, such as inverse reinforcement learning [Ng *et al.*, 2000; Evans *et al.*, 2016; Shah *et al.*, 2019; Hughes *et al.*, 2020; Zhi-Xuan *et al.*, 2020] that infers the reward functions in MDP through (potentially human) demonstrations. Researchers also aim to leverage human feedback to train reinforcement learning algorithms [Knox and Stone, 2008; Nikolaidis *et al.*, 2015; Bobu *et al.*, 2021]. Our work differs in that our goal is to induce humans to perform desired behavior through updating the decision-making environment instead of improving the learning algorithms. The more detailed discussion on the related work can be found in Appendix A.

2 Problem Setting

Decision-making environment. We formulate the sequential decision making environment as a finite-time horizon MDP with two sets of reward functions: $W = \langle S, A, P, R^a, R^p, T\rangle$, where S is the set of states, A is the set of agent actions, $P(s'|s,a)$ is the transition probability from state s to state s' after taking action a, T is the time horizon, $R^a(s,a)$ is the bounded reward obtained by the agent after he takes action a at state s, and $R^p(s,a)$ is the bounded reward obtained by the principal after the agent takes a at state s.

Agent decision-making policy. Since the agent could be biased and might not make time-consistent decisions, we represent the agent policy in a time-inconsistent manner: $\Pi : S \times T \to A$. In particular, for an agent policy $\pi \in \Pi$, $\pi(s,t)$ denotes the action the agent will take in state s at time t when following policy π. We formulate the agent as a planner $H : W \to \Pi$, with input being an environment $w \in W$ and output being a policy $\pi \in \Pi$ according to his decision-making model. The agent's goal is to maximize his *perceived* (potentially *biased*) rewards. To characterize the time-inconsistent behavior of the agent, we define the notion $d(t)$, the discounting factor that the agent perceives the payoff obtained t steps ahead. In the standard setting, $d(t)$ is often assumed to be in the form of γ^t with $\gamma \in (0,1]$ being the time-discounting factor. In this paper, we address different forms of $d(t)$ that captures different agent models, which will be discussed later.

With $d(t)$ defined, we now characterize the agent policy by defining a *perceived* Q-function[2] $Q^\pi(s,a,t,\hat{t})$, specifying the agent's perceived value at time t for him to take action a in state s at a future time $t+\hat{t}$ and follows policy π afterwards. This additional $\hat{t}$ parameter captures the agent's time-inconsistent belief: what the agent *thinks* he will do in a future time $t+\hat{t}$ while at time t might be different from

[2]This definition extends the standard Q-function to incorporate the agent's biased decision making.

what he will actually do at time $t+\hat{t}$. We also abuse the notation and let $\pi(s,t,\hat{t})$ denote the action the agent thinks what he would do in state s in a future time $t+\hat{t}$ while at time t. This perceived $Q^{\pi}(s,a,t,\hat{t})$ can be expressed as the sum of (1) the perceived reward for taking action a in a future time step $t+\hat{t}$ while at time t: $d(\hat{t})R^a(s,a)$ and (2) the expected future reward for following policy π after $t+\hat{t}$: $\mathbb{E}[\sum_{t'=t+\hat{t}+1}^{T} d(t'-t)R^a(s^{\pi}_{t'}, \pi(s^{\pi}_{t'}, t, t'-t))]]$, where $s^{\pi}_{t'}$ is the random variable denoting the state at t' if the agent follows π after $t+\hat{t}$. The expectation is over the randomness of the state transition.

Since the policy is only executed with $\hat{t}=0$ ($\hat{t}>0$ represents the agent's belief of what he would do $\hat{t}$ steps ahead), we let $Q^{\pi}(s,a,t) = Q^{\pi}(s,a,t,0)$ and $\pi(s,t) = \pi(s,t,0)$. The agent policy π^* can then be written as:

$$\pi^*(s,t) = \operatorname*{argmax}_a Q^{\pi^*}(s,a,t) \tag{1}$$

For a given environment, the agent policy can be solved by applying standard techniques, such as backward induction.

Biased agent models. As discussed above, we use the notion $d(t)$, denoting how much the agent discounts the payoff t steps in the future to characterize the agent's behavior. This notion characterizes many common behavioral models, with some illustrative examples below:

- Standard model: In the literature, the agent is often assumed to have a consistent time-discounting factor $\gamma \in (0,1]$ for discounting future payoff. Therefore, we can set $d(t)=\gamma^t$ to represent this standard assumption.
- Bounded rationality or short-sightedness: It considers the scenario in which the agent can only perform limited computation due to either time, cognitive, or information constraints. This can be approximated by considering that the agent only has information or only can reason about information within τ steps. We can formulate this by setting $d(t)=\gamma^t$ for all $0 \le t \le \tau$, and $d(t)=0$ for all $\tau < t \le T$. In the special case of *myopic agent*, who only cares about the immediate payoff and not the future payoffs, we can set $\tau = 0$.
- Present bias: When choosing between earning 10 dollars 100 days from now or 11 dollars 101 days from now, most people will choose the latter. However, when again being asked to choose between earning 10 dollars now or 11 dollars tomorrow, many people will change their decisions. This example illustrates the *present bias*, describing humans' inconsistency in discounting future payoffs. One common way to account for this behavior is through hyperbolic discounting factor: $d(t)=\frac{1}{1+kt}$ for $k>0$.

Design space of the principal. Recall that the principal aims to update the environment to influence the agent's actions. We consider two natural sets of "updates" the principal can make to the environment:

- Reward function modification: The principal may pay costs to modify the agent's reward function to influence the agent's decisions. Formally, the principal can modify the agent's reward from $R^a(s,a)$ to $\bar{R}^a(s,a) = R^a(s,a) + c(s,a)$ for taking action a in state s by paying a cost equal to the absolute value of the modification $|c(s,a)|$. The agent will only observe the modified reward function and will make decisions based on $\bar{R}^a$. Note that this type of environment updates is performed *offline* in the sense that it updates the environment before the agent starts to make their decisions in the environment.
- Action nudge: We also consider another design space, in which the principal can offer a non-negative incentive $c(s,a,t) \ge 0$ to *nudge* the agent to take action a in state s at time t. The agent's reward in state s would then be $R(s,a)+c(s,a,t)$ if taking action a at time t while the future perceived rewards do not change. Different from the reward function modification, this nudge influences the agent's decisions during *decision time*.

The principal's goal is to maximize her total rewards derived from the agent's actions under the budget constraint that the total cost does not exceed budget B. Given the agent's policy π and the initial state distribution $p_0(s)$, let $p^{\pi}_t(s)$ be the state distribution at time t when the agent follows policy π, the principal's total expected reward can be written as[3]:

$$\sum_{t=0}^{T}\sum_{s\in S} p^{\pi}_t(s)R^p(s,\pi(s,t)) \tag{2}$$

3 Problem Formulations and Algorithms

Before we formulate the environment design problems, we first present an important, although perhaps not surprising, result that if the agent exhibit biases in decision making, being oblivious of the biases could lead to undesired outcome for the principal. The result showcases the importance of taking human behavior into account in environment design[4].

Lemma 1. *If the principal performs environment design by assuming the agent is a standard agent while the agent is boundedly rational, the ratio between the principal's reward after environment design compared with the principal's reward obtained in environment design with the correct agent model could be arbitrarily close to* 0.

3.1 Reward function modification

We first consider the environment design problem in which the principal can influence the agent's decisions through modifying the agent's reward functions $R^a(s,a)$. Let $c(s,a)$ be the modification the principal makes on $R^a(s,a)$, and $\bar{R}^a(s,a) = R^a(s,a)+c(s,a)$ is the reward function that the agent perceives and based on when making decisions. Let the updated MDP environment be $\bar{w}$, replacing the agent reward function as $\bar{R}^a$, and the agent policy on this environment be $\pi = h(\bar{w})$. The environment design problem for the principal is to choose the set of updates $\{c(s,a)\}$ to maximize her payoff subject to the budget constraint B. Again, let the initial state distribution be $p_0(s)$, and $p^{\pi}_t(s)$ be the state distribution at time t when the agent follows policy π, we can formulate the environment design problem as follows,

[3] We do not include the time-discounting factor for the principal's payoff to simplify the presentations. Our results and discussion can be easily extended to the setting with time-discounting factor.

[4] All proofs are included in the appendix of the full paper.

$$\max_{c} \sum_{t=0}^{T} \sum_{s\in S} p_t^{\pi}(s) R^p(s, \pi(s,t)) \\ \text{s.t.} \sum_{s\in S} \sum_{a\in A} |c(s,a)| \le B \; ; \pi = h(\bar{w}) \tag{3}$$

Note that this is a bi-level optimization problem, in which the principal is optimizing over the space of $\{c(s,a)\}$ while the agent is optimizing his policy in response to the principal's update in the form of $\pi = h(\bar{w})$. To solve the inner optimization problem (the agent's optimal policy), we can define an updated $\bar{Q}^{\pi}$ by replacing the reward R^a with $\bar{R}^a$ and solve the policy π using backward induction. We show that this bi-level optimization problem is generally NP-hard to solve.

Theorem 2. *It is NP-hard to solve the environment design problem with reward function modification as defined in* (3)*.*

Relaxed formulation. To address this hardness result, we propose to use a soft-max stochastic policy ρ to relax the deterministic policy π. This relaxation makes the inner optimization differentiable, so first-order optimization methods might be applied. Instead of using $\pi(s,t)$ to denote the chosen action, we use $\rho(s,a,t)$ to represents the probability of choosing action a in state s at time t. Moreover, we again use $\bar{Q}^{\rho}$ to denote the perceived cumulative reward for policy ρ. The definition is similar to Q^{π} except that we need to incorporate the randomness of policy when evaluating the future reward. Moreover, we use a soft-max form to approximate the agent policy: $\rho(s,a,t) = \frac{e^{\beta \bar{Q}^{\rho}(s,a,t)}}{\sum_{a'} e^{\beta \bar{Q}^{\rho}(s,a',t)}}, \forall s,a,t$.

Below we formulate the relaxed environment design problem. We now use $p_t^{\rho}(s)$ to denote the state distribution at time t (with $p_0^{\rho}(s)$ defined as the initial state distribution $p_0(s)$ for notational simplicity) when the agent follows policy ρ. In addition, we explicitly layout the state distribution over time following policy ρ as a constraint in the third constraint of the optimization problem. Since the gradient of the optimization variables exists, we can approach this optimization through a gradient-based algorithm, as in Algorithm 1.

$$\max_{c} \sum_{t=0}^{T} \sum_{s\in S} \sum_{a\in A} p_t^{\rho}(s) R^p(s,a) \rho(s,a,t) \\ \text{s. t.} \sum_{s\in S} \sum_{a\in A} |c(s,a)| \le B \\ \rho(s,a,t) = \frac{e^{\beta \bar{Q}^{\rho}(s,a,t)}}{\sum_{a'} e^{\beta \bar{Q}^{\rho}(s,a',t)}}, \forall s,a,t \\ p_{t+1}^{\rho}(s) = \sum_{s'\in S} \sum_{a\in A} p_t^{\rho}(s') P(s|s',a) \rho(s',a,t), \forall s,t \\ \rho(s,a,t) \ge 0, \forall s,a,t \tag{4}$$

Discussion. When $\beta \to \infty$, $\rho(s,a,t)$ approximates to a delta function with the probability mass on the action with the highest $\bar{Q}$ value, which recovers the original problem. Moreover, recall that the Q function is defined with respect to the policy (when calculating the expected future rewards). We can show that this soft-max relaxation converges to the Q function of deterministic policy exponentially fast in β. In our simulations, we also empirically demonstrate that setting a small β is enough to approximate the optimal of the original problem in (3).

Algorithm 1 Gradient-based Algorithm for Solving (4)

Input: learning rate δ, maximal iterations N
initialize $c, i = 0$
while $i < N$ **do**
  sample $\hat{s} \in S, \hat{a} \in A$
  update $\bar{R}^a(s,a), \bar{Q}(s,a,t), \rho(s,a,t), p_t^{\rho}(s), \forall s,a,t$
  calculate $\frac{\partial \rho(s,a,t)}{\partial c(\hat{s},\hat{a})}, \frac{\partial p_t^{\rho}(s)}{\partial c(\hat{s},\hat{a})}, \forall s,a,t$
  $c(\hat{s},\hat{a}) \leftarrow c(\hat{s},\hat{a}) + \delta \frac{\partial \sum p_t^{\rho}(s) R^p(s,a) \rho(s,a,t)}{\partial c(\hat{s},\hat{a})}$
  $i \leftarrow i + 1$
end while
return c

Lemma 3. *For any environment w, let π_w and ρ_w be the agent's deterministic and stochastic policies following our model. Let $Q^{\pi_w}(s,a,t)$ and $Q^{\rho_w}(s,a,t)$ be the corresponding Q-functions. For all (s,a,t), we have*

$$|Q^{\pi_w}(s,a,t) - Q^{\rho_w}(s,a,t)| \le \mathcal{O}(e^{-\beta C}),$$

where $C > 0$ is a constant and β is the parameter of ρ.

3.2 Action nudge

We now formulate the environment design problem via action nudge. The principal can choose to pay $c(s,a,t) \ge 0$ to the agent if he takes action a in state s at time t. In this approach, the agent's perceived Q function does not change, but the agent's action will be influenced by this additional incentive, i.e., the agent will choose the action that maximizes $Q^{\pi}(s,a,t) + c(s,a,t)$ in state s at time t. Moreover, since the nudge is calculated offline but deployed online, the budget constraint is satisfied in expectation. Formally, the principal's environment design problem can be written as:

$$\max_{c} \sum_{t=0}^{T} \sum_{s\in S} p_t^{\pi}(s) R^p(s, \pi(s,t)) \\ s.t. \sum_{t=0}^{T} \sum_{s\in S} c(s, \pi(s,t), t) p_t^{\pi}(s) \le B \\ \pi(s,t) = \operatorname*{argmax}_{a} \{Q^{\pi}(s,a,t) + c(s,a,t)\}, \forall s,t \tag{5}$$

Solving this problem directly is again generally NP-hard due to the same bi-level optimization property and the deterministic policy structure. Below we utilize the problem structure and develop an alternative formulation.

Alternative formulation. Let π be the agent's policy in the original decision-making environment. The goal of action nudge is to make the agent change from action $a = \pi(s,t)$ to a new action a'. Assume the principal can break ties in any way she prefers when multiple actions lead to the same payoff[5], the cost the principal needs to pay to make the agent select action a' instead of a is $c(s,a',t) = Q(s,a,t) - Q(s,a',t)$.

[5]While this assumption seems strong, it can be approximately satisfied by adding a arbitrarily small value to $c(s,a',t)$ to make the agent break ties to align with the principal's goal.

We can pre-calculate all the cost the principal needs to pay for action nudge $c(s,a,t) = Q(s,\pi(s,t),t) - Q(s,a,t), \forall s,a,t$.

With the above observations and the additional tie-breaking assumption, the environment design problem via action nudge is reduced to selecting which action the principal should nudge the agent to select for all (s,t). The nudged action a would generate a reward of $R^p(s,a)$ and incurs a cost $c(s,a,t)$. The goal is to maximize the total rewards such that the total cost is no larger than budget B in expectation. This problem reduces to a standard constrained MDP problem.

$$\begin{aligned} &\max_{\phi} \sum_{t=0}^{T} \sum_{s\in S} \sum_{a\in A} R^p(s,a)\phi(s,a,t) \\ s.t. &\sum_{t=0}^{T} \sum_{s\in S} \sum_{a\in A} c(s,a,t)\phi(s,a,t) \le B \\ &\sum_{s'\in S} \sum_{a\in A} P(s|s',a)\phi(s',a,t) = \sum_{a\in A} \phi(s,a,t+1), \forall s,t \\ &\sum_{a\in A} \phi(s,a,0) = p_0(s), \forall s \\ &\phi(s,a,t) \ge 0, \forall s,a,t \end{aligned} \tag{6}$$

In this optimization problem, $\phi(s,a,t)$ is the optimization variables, representing the joint probability at time t for the agent to be in state s and take action a. To translate $\phi(s,a,t)$ to the stochastic policy $\rho(s,a,t)$, we have $\rho(s,a,t) = \frac{\phi(s,a,t)}{\sum_{a'\in A}\phi(s,a',t)}$. The optimization problem is a linear program in $\phi(s,a,t)$. Therefore we can directly apply standard linear programming solvers to solve this optimization problem. When the agent is in state s at time t, this solution indicates that the principal should nudge and offers $c(s,a,t)$ if $\phi(s,a,t) > 0$. [6]

4 Experiments

We conduct both simulated and real-human experiments to evaluate our proposed algorithms for environment design.

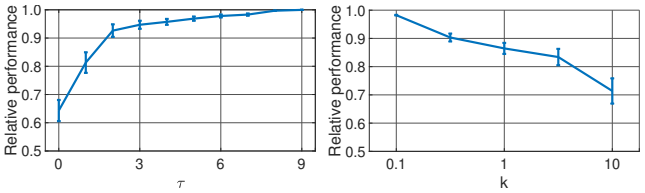
4.1 Simulations

In our simulations, we create a grid world of size 10×10. Each grid represents a state in the MDP. There are four actions representing the direction agent can move to: {up, down, left, right}. After each action, the agent moves to the nearby grid associated with the action with 70% chance and to a random nearby grid with 30% chance. The initial state is in the middle of the grid world. The time horizon T is set to be 20.

We initialize the principal's reward function values to be uniformly drawn from the range $[0, 0.5]$. We then randomly choose a 2×2 block as global optimal region and add 0.5 to the reward values within this block. Similarly, we randomly draw 1 to 3 local optimal regions (2×2 blocks) by setting their reward lower than global optimal but higher than its neighbors. We randomly generate 1,000 environments following the above procedure and report the average results on these 1,000 environments.

Different agent behavioral models. We start with the setting that the agent's reward function is the same as the principal's. In this setting, if the agent is behaving optimally, the principal does not need to update the environment. Therefore, we focus on examining how the agent's biased behavior impacts the total payoff and how effectively environment design can help.

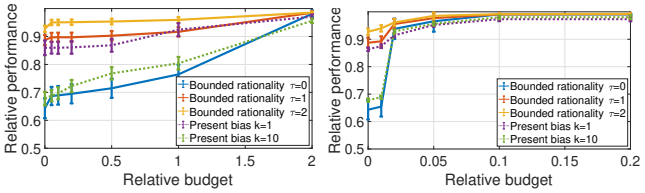
We first examine the impact of biased agents without environment design. We consider agents with bounded rationality (or short-sightedness) and with present bias. Following the formulation in Section 2, we modify τ for boundedly-rational agents and k for present-bias agents. For boundedly-rational agents, we set $\gamma = 1$ and vary τ to be from 0 to 9. For present-bias agents, we vary k to be in $\{0.1, \sqrt{0.1}, 1, \sqrt{10}, 10\}$. The performance is measured in terms of the principal's objective. As shown in Figure 1, the principal's payoff, even when the reward function aligns with the agent's, could decrease significantly when the agent exhibits decision biases.



(a) Bounded rationality. (b) Present biased.

Figure 1: The principal's payoff with biased decision-makers without environment design. Agents with higher τ or lower k are closer to being rational.

Next we examine the effect of environment design in improving the principal's payoff. We apply the algorithms in Section 3, with the soft-max parameter $\beta = 3$ (the choice of β is discussed in the appendix). We examine present-bias agents with $k \in \{1, 10\}$ and boundedly-rational agents with $\tau \in \{0, 1, 2\}$. We vary the budget for algorithms with both design spaces. As in Figure 2, our algorithms lead to effective environment design and improve with larger budget. While action nudge seems more cost efficient, the cost needs to be incurred for each agent. In reward modification, the environment may need only be updated once for multiple agents.



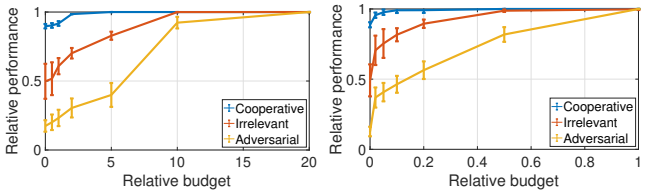
(a) Reward modification. (b) Action nudge.

Figure 2: The principal's payoff with biased decision-makers after applying environment design. The y-axis is the relative performance compared with the optimal, and the x-axis is the amount of budget relative to the optimal performance.

Mis-alignment of the principal's and the agent's objective. We now consider the case that the agent's reward function might not align with the principal's. We fix the principal's reward function as before and vary the agent's reward function. We consider the cases in which the agent's reward function

[6] There could be multiple actions that lead to $\phi(s,a,t) > 0$ for a given (s,t), leading to offering multiple nudges simultaneously. In Appendix C, we show that there exists a solution such that this does not happen frequently and discuss approaches to find this solution.

is the inverse (adversarial), randomly drawn (irrelevant), and the same (cooperative) of the principal's reward function. The agent's bias model is set to be boundedly rational with $\tau = 1$ (the results are qualitatively similar for other agent models). As shown in Figure 3, our algorithm can find the sets of environment updates to induce desired agent decisions, though it generally requires more budgets when the principal's reward function does not align with the agent's.



(a) Reward modification. (b) Action nudge.

Figure 3: Misalignment of the principal's and the agent's the agent's reward function. The y-axis is the relative performance compared with the optimal (in terms of the principal's payoff), and the x-axis is the amount of budget relative to the optimal performance.

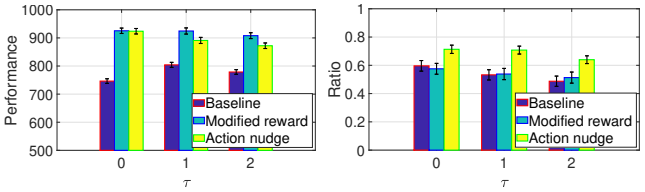
Additional simulations. Additional simulations are included in Appendix D. We show that setting a small β in Algorithm 1 suffices to approximate the true optimal of (3) and examine its runtime. This result complements Lemma 3, proving the convergence of Q functions, and demonstrates that we can approximate the overall performance of the optimal. In another simulation, we demonstrate how to combine off-the-shelf inverse reinforcement learning algorithms to deal with scenarios when the agent rewards and biases are unknown a priori.

4.2 Real-world human-subject experiments

While our simulation results are promising, they are under the assumption that the agent makes decisions following the behavioral model. In this section, we examine whether our environment design algorithms are effective for real human decision makers whose behavior might deviate from the model. We have recruited 300 unique workers from Amazon Mechanical Turk. Each worker is paid \$0.50 and might earn additional bonuses. The average hourly rate is around \$11.50.

Task description. Each worker is asked to play six navigation games, with each represented by a grid world of size 10×10. The setup is similar to our simulations, except that we simplify the rewards to depend only on the state, i.e., $R^a(s,a) = R^p(s,a) = R(s)$, to reduce the cognitive burden for workers. Workers' bonuses depend on their total rewards. We also consider the setting in which the principal and the agent share the same reward function. To induce biased human behavior, a worker can only see the rewards of the nearby states (to simulate the short-sightedness). Out of six games, there are two games each for vision length of $1, 2, 3$, which we use short-sighted (boundedly rational) agent with $\tau = 0, 1, 2$ to model when solving the environment design problem. The detailed task setup is included in Appendix E.1.

Each worker is randomly assigned to one of the three treatments: {baseline, modified reward, action nudged}. The games are drawn from the same pool for each treatment. In baseline, workers play the drawn games without modifications. In modified reward, workers see the modified rewards generated by our algorithm. In action nudge, when a nudge happens, the workers see an additional messages indicating they might gain bonus for moving towards a certain direction. Since our goal is to observe whether environment design has impacts to real human decision-makers, we set the budget to be large enough such that the optimal decisions can be induced when the agent follows the behavioral model. We also report the true incurred cost in the experiment results.



(a) Average principal's payoff. (b) The ratio of human moves matched predictions.

Figure 4: The results from the human-subject experiment. The results are grouped by the vision length of the games, mapping to different values of τ in short-sighted (boundedly rational) agents. 4(a) shows the average principal's payoff with real human decision makers in treatments, and 4(b) shows the ratio of worker moves which are the same as short-sighted model predictions.

Experiment results. As shown in Figure 4(a), workers under both environment design treatments generate more rewards for the principal, suggesting that our algorithms lead to effective environment designs even for real humans that do not always behave as the behavioral model. The actual costs incurred in "modified reward" and "action nudge" treatments are 73.7 and 50.3 points, while the average gain is 142.9 and 119.2 points. Moreover, since the principal and the agent share the same reward, the baseline treatment corresponds to the optimal design (do nothing) for the standard agent model. The performance improvement of our algorithms re-affirms the importance of incorporating realistic human models.

We also measure whether real humans behave as predicted by the behavioral model. As in Figure 4(b), worker behavior aligns with our behavioral models 53.8%, 54.2%, 68.7% of the time on average in each treatment. We also compare worker behavior with the standard model, with alignment at only 33.2%, 36.9%, 45.9% of the time. Interestingly, workers are more likely to behave as predicted in the "action nudge" treatment, likely because this treatment generates additional information that triggers workers to follow the nudged action.

5 Conclusion

We investigate environment design with biased decision makers. Our work sheds lights on many important applications, such as AI-assisted decision making. Future works include incorporating other bias models, including different environment design strategies, and addressing potential concerns when the objectives of the principal and the agent differ, such as in the adversarial setting. For example, can we design robust decision-making environments, e.g., imposing regulations/constraints on the environment updates to be allowed, to better safeguard human welfare. We hope this work can open more discussion in designing assistive AI technology and in incorporating behavioral models in computation.

Acknowledgements

This work is supported in part by the Office of Naval Research Grant N00014-20-1-2240.

References

Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *International Conference on Machine Learning*, 2004.

Andreea Bobu, Marius Wiggert, Claire Tomlin, and Anca D Dragan. Feature expansive reward learning: Rethinking human input. In *International Conference on Human-Robot Interaction*, 2021.

Xiaoni Duan, Chien-Ju Ho, and Ming Yin. Does exposure to diverse perspectives mitigate biases in crowdwork? an explorative study. In *AAAI Conference on Human Computation and Crowdsourcing*, 2020.

Xiaoni Duan, Chien-Ju Ho, and Ming Yin. The influences of task design on crowdsourced judgement: A case study of recidivism risk evaluation. In *The Web Conference (WWW)*, 2022.

Owain Evans and Noah D Goodman. Learning the preferences of bounded agents. In *NIPS Workshop on Bounded Optimality*, 2015.

Owain Evans, Andreas Stuhlmüller, and Noah Goodman. Learning the preferences of ignorant, inconsistent agents. In *AAAI Conference on Artificial Intelligence*, 2016.

Peter Frazier, David Kempe, Jon Kleinberg, and Robert Kleinberg. Incentivizing exploration. In *ACM Conference on Economics and Computation*, 2014.

Sebastian Gottwald and Daniel A Braun. Bounded rational decision-making from elementary computations that reduce uncertainty. *Entropy*, 2019.

Dana Hughes, Akshat Agarwal, Yue Guo, and Katia Sycara. Inferring non-stationary human preferences for human-agent teams. In *IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2020.

Daniel Kahneman. A perspective on judgment and choice: mapping bounded rationality. *American Psychologist*, 58(9):697, 2003.

Jon Kleinberg and Sigal Oren. Time-inconsistent planning: a computational problem in behavioral economics. In *ACM Conference on Economics and Computation*, 2014.

Jon Kleinberg, Sigal Oren, and Manish Raghavan. Planning with multiple biases. In *ACM Conference on Economics and Computation*, 2017.

W. Bradley Knox and Peter Stone. Tamer: Training an agent manually via evaluative reinforcement. In *IEEE International Conference on Development and Learning*, 2008.

Yishay Mansour, Aleksandrs Slivkins, and Vasilis Syrgkanis. Bayesian incentive-compatible bandit exploration. In *ACM Conference on Economics and Computation*, 2015.

Peta Masters, Michael Kirley, and Wally Smith. Extended goal recognition: a planning-based model for strategic deception. In *International Conference on Autonomous Agents and MultiAgent Systems*, 2021.

Peta Masters, Wally Smith, and Michael Kirley. Extended goal recognition: Lessons from magic. *Frontiers in Artificial Intelligence*, 4, 2021.

Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *International Conference on Machine Learning*, 2000.

Stefanos Nikolaidis, Ramya Ramakrishnan, Keren Gu, and Julie Shah. Efficient model learning from joint-action demonstrations for human-robot collaborative tasks. In *International Conference on Human-Robot Interaction*, 2015.

Ted O'Donoghue and Matthew Rabin. Doing it now or later. *American Economic Review*, 89(1):103–124, 1999.

Amin Rakhsha, Goran Radanovic, Rati Devidze, Xiaojin Zhu, and Adish Singla. Policy teaching via environment poisoning: Training-time adversarial attacks against reinforcement learning. In *International Conference on Machine Learning*, 2020.

Deepak Ramachandran and Eyal Amir. Bayesian inverse reinforcement learning. In *International Joint Conference on Artificial Intelligence*, 2007.

Sonja Schach, Sebastian Gottwald, and Daniel A Braun. Quantifying motor task performance by bounded rational decision theory. *Frontiers in neuroscience*, 2018.

Rohin Shah, Noah Gundotra, Pieter Abbeel, and Anca Dragan. On the feasibility of learning, rather than assuming, human biases for reward inference. In *International Conference on Machine Learning*, 2019.

Aleksandrs Slivkins. Introduction to multi-armed bandits. *Found. Trends Mach. Learn.*, 2019.

Zhao Song, Ron Parr, and Lawrence Carin. Revisiting the softmax bellman operator: New benefits and new perspective. In *International Conference on Machine Learning*, 2019.

Wei Tang and Chien-Ju Ho. Bandit learning with biased human feedback. In *International Conference on Autonomous Agents and Multiagent Systems*, 2019.

Wei Tang and Chien-Ju Ho. On the bayesian rational assumption in information design. In *AAAI Conference on Human Computation and Crowdsourcing*, 2021.

Wei Tang, Chien-Ju Ho, and Ming Yin. Leveraging peer communication to enhance crowdsourcing. In *The Web Conference (WWW)*, 2019.

Shunan Zhang and J Yu Angela. Forgetful bayes and myopic planning: Human learning and decision-making in a bandit setting. In *Advances in Neural Information Processing Systems*, 2013.

Haoqi Zhang and David C Parkes. Value-based policy teaching with active indirect elicitation. In *AAAI Conference on Artificial Intelligence*, 2008.

Haoqi Zhang, Yiling Chen, and David C Parkes. A general approach to environment design with one agent. In *International Joint Conference on Artificial Intelligence*, 2009.

Haifeng Zhang, Jun Wang, Zhiming Zhou, Weinan Zhang, Ying Wen, Yong Yu, and Wenxin Li. Learning to design games: Strategic environments in reinforcement learning. In *International Joint Conference on Artificial Intelligence*, 2018.

Xuezhou Zhang, Yuzhe Ma, Adish Singla, and Xiaojin Zhu. Adaptive reward-poisoning attacks against reinforcement learning. In *International Conference on Machine Learning*, 2020.

Tan Zhi-Xuan, Jordyn Mann, Tom Silver, Josh Tenenbaum, and Vikash Mansinghka. Online bayesian goal inference for boundedly rational planning agents. In *Advances in Neural Information Processing Systems*, 2020.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *AAAI Conference on Artificial Intelligence*, 2008.

A Related Work

Our work is built on the formulation of Markov decision process commonly seen in reinforcement learning. Instead of solving the agent's optimal policy, in this work, we consider a Stackelberg game formulation, in which the principal can first choose how to update the decision-making environment, and then the agent makes decisions in the updated environment. The goal of the principal is to obtain the maximum total rewards derived from the agent's actions. When the agent is rational and chooses the optimal policy, our problem is similar to policy teaching [Zhang and Parkes, 2008; Zhang *et al.*, 2009, 2018] and poisoning attack for reinforcement learning [Rakhsha *et al.*, 2020; Zhang *et al.*, 2020] in the literature. Our work deviates from these works by incorporating human behavioral models in the framework and in conducting real-world human subject experiments to evaluate our approaches.

This work incorporates the human behavioral models about biased decision-making from behavioral economics. In particular, we include the *bounded rationality* [Kahneman, 2003], which describes the intuitions that human decisions might not be optimal due to limited computation power or lack of future information (a myopic agent can be considered as a bounded-rational agent that only cares about the current payoff), and *present bias* [O'Donoghue and Rabin, 1999], which describes humans' tendency to give stronger weights on immediate costs and benefits rather than balancing them against costs and benefits in the future. While these behavioral models are empirically observed to often better align with real human behavior, there are still relatively limited research that incorporate them in studying computational systems with humans in the loop.

There have been recent works that aim to incorporate behavioral models in computational frameworks. For example, the research on incentivizing exploration [Frazier *et al.*, 2014; Mansour *et al.*, 2015] (see Chapter 11 in the recent survey by Slivkins [2019]) studies how a principal can incentivize myopic agents to perform exploration in bandit learning via designing specific monetary payments or information policies; Tang and Ho [2019] incorporates the model of herding bias in the feedback generation in bandit learning; Tang and Ho [2021] relax the Bayesian rational assumption in incentive design; Kleinberg and Oren [2014] and Kleinberg *et al.* [2017] study the planning for time-inconsistent agents in environments characterized by graphical models; Masters *et al.* [2021a,b] incorporate biased human model in goal recognition. Moreover, there have been works examining real-world human behavior in computational environments [Zhang and Angela, 2013; Schach *et al.*, 2018; Gottwald and Braun, 2019; Duan *et al.*, 2020, 2022; Tang *et al.*, 2019]. Our work aligns with this line of research which incorporates realistic human behavioral models in computation. In particular, we focus on the study of environment design in sequential decision-making environments characterized by Markov decision process with biased decision makers.

There have been other lines of research that also includes humans in the loop of reinforcement learning frameworks. For example, inverse reinforcement learning [Ng *et al.*, 2000; Abbeel and Ng, 2004; Ziebart *et al.*, 2008; Ramachandran and Amir, 2007] aims to infer the reward functions in MDP through observing demonstrations of the optimal policy. If the demonstrator is a human being, the demonstrations could be noisy or contain behavioral biases. There have been studies [Evans *et al.*, 2016; Shah *et al.*, 2019; Hughes *et al.*, 2020; Zhi-Xuan *et al.*, 2020] aiming to incorporate human behavioral biases in the inference process and infer both the rewards and biases simultaneously. While the research goal is different, this line of research complements our study in that the techniques can be applied to infer the reward function and human biases in our formulation. In another line of research, researchers aim to leverage human feedback and knowledge to better train learning algorithms in reinforcement learning settings [Knox and Stone, 2008; Nikolaidis *et al.*, 2015; Bobu *et al.*, 2021]. This work differs from this line of work in that our goal is to induce humans to perform desired behavior through finding optimal ways to update the decision-making environment instead of improving the learning algorithms.

B Omitted Proofs in the Main Paper

B.1 Proof of Lemma 1

Proof. We prove the lemma by constructing an example MDP for bounded-rational agents. Consider a bounded-rational agent with parameter τ. We construct an MDP as show in Figure A1, where the circle denotes the state, arrow denotes the action (with deterministic transition), and the number associate with the arrow is the reward. We consider the case that the reward functions for the principal and the agent are the same. In this example, the set of state is $\{s_0, ..., s_{\tau+1}\}$. For states s_i with $i = 1$ to τ, there is only one available action "move right" that moves to state s_{i+1}, where the reward $R^a(s_i, \text{move right}) = R^p(s_i, \text{move right}) = 1$. For state s_0, there is an additional action of staying in state s_0 that lead to reward of 2, and for state $s_{\tau+1}$, the only action is to move to state s_0 that gives a reward of m.

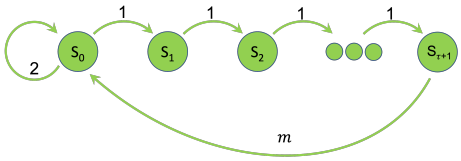


Figure A1: The example MDP used for proving Lemma 1 with bounded-rational agents.

Let the initial state be s_0 and $T = \tau + 2$. If the agent is bounded rational with τ, it is easy to see that the agent will choose to stay in s_0 and generate a total reward of $(\tau + 2) * 2$ for the principal. If the agent is a standard agent with $\gamma = 1$, he will move from s_0 to $s_{\tau+1}$ then back to s_0, leading to a total reward of $\tau + 1 + m$. Therefore, without environment design, the ratio between the reward by a bounded-rational agent and the reward by a standard agent would be $2(\tau + 2)/(\tau + 1 + m)$, which goes to 0 when $m \to \infty$. This proves that there exists an MDP such that the ratio of the reward made by an bounded rational agent compared to the reward made by a standard agent is arbitrarily close to 0.

If the principal believes the agent is a standard agent, she does not need to update the environment to reach the optimal payoff. However, the agent, who is bounded rational, would take the the sub-optimal action. Therefore, the principal's performance ratio would again be $2(\tau+2)/(\tau+1+m)$, which could goes to 0 when $m \to \infty$. □

B.2 Proof of Theorem 2

Proof. We prove the NP-hardness through a reduction from the knapsack problem. The reduction process is similar to the one by Zhang and Parkes [2008], though we use the myopic agent case to show that the problem is NP-hard with biased agents. Note that this proof holds for both the problem of environment design via reward function modification and via action nudge.

An instance of the knapsack problem consists of n items. Denote the i-th item value as $u_i > 0$ and the weight as $w_i > 0$, for $1 \le i \le n$. The knapsack problem aims to find a set of items that maximizes the total values while ensuring the total weights is within budget B. Using variables $x_i \in \{0,1\}$ to denote whether item i is included in the set, the knapsack problem can be formulated as the below integer program, which is NP-hard in general:

$$\max_x \sum_{i=1}^n u_i x_i;\ \text{s.t.} \quad \sum_{i=1}^n w_i x_i \le B,\ x_i \in \{0,1\}, \forall i \tag{A1}$$

For the reduction, for each instance of the knapsack problem, we can construct an MDP as follows. Consider an MDP with $n+1$ states, s_1 to s_{n+1}. The initial state is s_1. We also make the action set from state s_{n+1} to be empty, effectively making it an end state. There are two actions $\{a_1, a_0\}$ available for each state, except for s_{n+1}, with taking a_1 at state s_i representing accepting item i, and taking a_0 representing not accepting. No matter which action is taken at state s_i, the next state will always be s_{i+1}. The agent's reward function corresponds to the weight for accepting item and the principal's rewards corresponds to the utility, i.e., agent reward is $R^a(s_i, a_1) = -w_i$, and principal reward is $R^p(s_i, a_1) = u_i$. Taking action a_0 leads to zero reward for both principal and agent, i.e., $R^a(s_i, a_0) = R^p(s_i, a_0) = 0$. Let the agent be myopic, and the budget for the principal is B. Note that in this MDP, the initial state is s_1, and no matter what the agent policy is, the state at time t is s_t, so we have omitted the time index in the agent policy to simplify the presentation, which means the myopic agent will choose $\pi(s_i) = \text{argmax}_{a\in\{a_0,a_1\}} R^a(s_i, a)$. Note also that, without environment design, the myopic agent will always choose action a_0 since $R^a(s_i, a_0) = 0 > -w_i = R^a(s_i, a_1), \forall i$, and therefore the principal's total reward is 0 in this case.

The environment design problem either via reward function modification or action nudge can be expressed in the same way as in (A2). The reason is that 1) in this MDP construction, the agent policy does not depend on time t, and 2) since the agent is myopic, both reward function modification and action nudge need to pay same amount of $c(s_i, a)$ for agent to change action. Therefore, the optimization formulation for both design spaces is the same as in (A2), and this NP-hardness proof holds for both cases.

$$\begin{aligned} &\max_c \sum_{i=1}^n R^p(s_i, \pi(s_i)) \\ &\text{s.t.} \sum_{i=1}^n \sum_{a\in\{a_0,a_1\}} |c(s_i, a)| \le B \\ &\pi(s_i) = \operatorname*{argmax}_a \{R^a(s_i, a) + c(s_i, a)\}, \forall s_i \end{aligned} \tag{A2}$$

Below we show that, with the solution of (A2), we can obtain the solution of the knapsack problems in polynomial time. Since we can construct the environment design problem for every instance of the knapsack problem, if our environment design problem is not NP-hard, the knapsack problem is not NP-hard, which lead to the contradiction since the knapsack problem is known to be NP-hard. Observe that if we have the solution $c(s, a)$ from (A2), we can obtain $\pi(s_i)$ as well in the equality constraint. If $\pi(s_i) = a_0$, we have $c(s_i, a_0) = c(s_i, a_1) = 0$. If $\pi(s_i) = a_1$, we have $c(s_i, a_1) - c(s_i, a_0) \ge R^a(s_i, a_0) - R^a(s_i, a_1) = w_i$. Therefore, $|c(s_i, a_1)| + |c(s_i, a_0)| \ge w_i$, and the equality holds when we set $c(s_i, a_1) = w_i$ and $c(s_i, a_0) = 0$. With the above observation and our MDP construction, setting x_i to be 1 in the knapsack problem if and only if $\pi(s_i) = a_1$ could maximize the total utility of selected items while satisfying the budget constraint on item weights. This means we can solve the knapsack problem if the solution of (A2) is given. This finishes the proof. □

B.3 Proof of Lemma 3

Recall that we define $Q(s, a, t)$ as $Q(s, a, t, 0)$. Below we give the proof of a more general version of Lemma 3, as stated below.

Lemma A1. *For any environment w, let π_w and ρ_w be the agent's deterministic and stochastic policies following our model. Let $Q^{\pi_w}(s, a, t, \hat{t})$ and $Q^{\rho_w}(s, a, t, \hat{t})$ be the corresponding Q-functions. For all $(s, a, t, \hat{t})$, we have*

$$|Q^{\pi_w}(s, a, t, \hat{t}) - Q^{\rho_w}(s, a, t, \hat{t})| \le \mathcal{O}(e^{-\beta C}),$$

where $C > 0$ is a constant and β is the parameter of ρ.

Proof. This proof extends the results by Song *et al.* [2019], who prove the convergence for infinite-time horizon MDP, to address finite horizon and general discounting function. In the following proof, we omit the subscript w in π_w and ρ_w and represent them using π and ρ. For a biased agent with discounting function $d(t)$, Let $Q^\pi(s, a, t, \hat{t})$ be the biased Q-function following π. Similar to the standard notation convention, we use the random variable s_t^π to denote state at time t when following policy π. The expectation is taken over the randomness of state transition. Also, since we are considering finite-horizon MDP, we set $Q^\pi(s, a, t, \hat{t}) = 0$ for $t + \hat{t} > T$, which represents the unreachable horizon. We have

$$\begin{aligned} Q^\pi(s, a, t, \hat{t}) &= d(\hat{t}) R(s, a) \\ &\quad + \mathbb{E}[Q^\pi(s^\pi_{t+\hat{t}+1}, \pi(s^\pi_{t+\hat{t}+1}, t, \hat{t}+1), t, \hat{t}+1)] \end{aligned}$$

We can write down $Q^\rho(s,a,t,\hat{t})$ similarly as Q^π. The only difference is in the second term. Instead of taking action $\pi(s^\pi_{t+\hat{t}+1}, t, \hat{t}+1)$, the agent takes action $a = a^\rho_{t+\hat{t}+1}$ with probability $\rho(s,a,t,\hat{t}+1) = \frac{e^{\beta Q^\rho(s,a,t,\hat{t}+1)}}{\sum_{a'} e^{\beta Q^\rho(s,a',t,\hat{t}+1)}}$. Moreover, the expectation is taken over both state transition and stochastic policy.

$$Q^\rho(s,a,t,\hat{t}) = d(\hat{t})R(s,a) + \mathbb{E}[Q^\rho(s^\rho_{t+\hat{t}+1}, a^\rho_{t+\hat{t}+1}, t, \hat{t}+1)]$$

Claim 1: $Q^\pi(s,a,t,\hat{t}) - Q^\rho(s,a,t,\hat{t}) \geq 0$

Proof. We prove this claim by induction. Note that by definition, both Q functions are 0 when $\hat{t}+t > T$.

- When $\hat{t} = T-t$, $Q^\pi(s,a,t,T-t) - Q^\rho(s,a,t,T-t) = R(s,a) - R(s,a) = 0$.
- When $\hat{t} < T-t$, we have $Q^\pi(s,a,t,\hat{t}-1) - Q^\rho(s,a,t,\hat{t}-1) = \mathbb{E}[\max_a Q^\pi(s,a,t,\hat{t}) - \sum_a \rho(s,a,t,\hat{t})Q^\rho(s,a,t,\hat{t})]$. Since $Q^\pi(s,a,t,\hat{t}) \geq Q^\rho(s,a,t,\hat{t})$ for all $(s,a,t,\hat{t})$, we have $Q^\pi(s,a,t,\hat{t}) - Q^\rho(s,a,t,\hat{t}) \geq 0$.

□

For the purpose of the analysis, we define two functions:

$$\delta(s,t,\hat{t}) = \max_a Q^\pi(s,a,t,\hat{t}) - \sum_a \rho(s,a,t,\hat{t})Q^\pi(s,a,t,\hat{t})$$

$$\zeta(t,\hat{t}) = \max_s \delta(s,t,\hat{t})$$

Claim 2: $Q^\pi(s,a,t,\hat{t}) - Q^\rho(s,a,t,\hat{t}) \leq \sum_{j=t+\hat{t}+1}^{T} \zeta(t, j-t)$

Proof. We again prove it by induction.

- When $\hat{t} = T-t$, $Q^\pi(s,a,t,T-t) - Q^\rho(s,a,t,T-t) = R(s,a) - R(s,a) = 0$.
- Suppose the statement is true for $\hat{t}$. For $\hat{t}-1$, we have (for notation simplicity, we use s' to denote $s_{t+\tau}$). The expectation of s' is over the state transition $P(s'|s,a)$ and the expectation of a' is over the stochastic policy ρ.

$$\begin{aligned}
&Q^\pi(s,a,t,\hat{t}-1) - Q^\rho(s,a,t,\hat{t}-1) \\
=&\mathop{\mathbb{E}}_{s'}\big[\max_{a'} Q^\pi(s',a',t,\hat{t}) - \mathop{\mathbb{E}}_{a'}[Q^\rho(s',a',t,\hat{t})]\big] \\
\leq&\mathop{\mathbb{E}}_{s'}\big[\max_{a'} Q^\pi(s',a',t,\hat{t}) - \mathop{\mathbb{E}}_{a'}[Q^\pi(s',a',t,\hat{t})] \\
&+ \sum_{j=t+\hat{t}+1}^{T} \zeta(t,j-t)\big] \\
=&\mathop{\mathbb{E}}_{s'}[\delta(s',t,\hat{t})] + \sum_{j=t+\hat{t}+1}^{T} \zeta(t,j-t) \\
\leq&\zeta(t,\hat{t}) + \sum_{j=t+\hat{t}+1}^{T} \zeta(t,j-t) = \sum_{j=t+\hat{t}}^{T} \zeta(t,j-t)
\end{aligned}$$

Therefore, by induction, we know the claim is true. □

With the above claims, we now show how ζ converges in terms of β. For given $(s,t,\hat{t})$, we sort $\{Q^\pi(s,a,t,\hat{t})\}$ such that $Q^\pi(s,a_{[1]},t,\hat{t}) \geq Q^\pi(s,a_{[2]},t,\hat{t}) \geq \cdots \geq Q^\pi(s,a_{[m]},t,\hat{t})$. Therefore, we have $\sigma_i = Q^\pi(s,a_{[1]},t,\hat{t}) - Q^\pi(s,a_{[i]},t,\hat{t}) \geq 0$. Also, there exists an index $i^* \leq m$ such that $\sigma_i > 0, \forall i^* \leq i \leq m$ and $\sigma_i = 0, \forall i < i^*$. If i^* does not exist, for all action $Q^\pi(s,a,t,\hat{t}) = \max_{a'} Q^\pi(s,a',t,\hat{t})$, there is no difference in selecting any action. Notice that i^* depends on $(s,t,\hat{t})$, but we omit the dependency for clarity.

Note that we can express $\delta(s,t,\hat{t})$ as below.

$$\begin{aligned}
\delta(s,t,\hat{t}) =& Q^\pi(s,a_{[1]},t,\hat{t}) - \sum_a \rho(s,a,t,\hat{t})Q^\pi(s,a,t,\hat{t}) \\
=&\frac{\sum_{i=2}^m e^{-\beta\sigma_i}\sigma_i}{1+\sum_{i=2}^m e^{-\beta\sigma_i}}
\end{aligned}$$

Since $\frac{\sum_i x_i}{1+\sum_i y_i} \leq \sum_i \frac{x_i}{1+y_i}$ for non-negative sequences $\{x_i\}$ and $\{y_i\}$. By setting $x_i = e^{-\beta\sigma_i}\sigma_i$ and $y_i = e^{-\beta\sigma_i}$, we have

$$\begin{aligned}
\delta(s,t,\hat{t}) =& \frac{\sum_{i=2}^m e^{-\beta\sigma_i}\sigma_i}{1+\sum_{i=2}^m e^{-\beta\sigma_i}} \\
\leq& \sum_{i=2}^m \frac{e^{-\beta\sigma_i}\sigma_i}{1+e^{-\beta\sigma_i}} \\
=& \sum_{i=2}^m \frac{\sigma_i}{1+e^{\beta\sigma_i}} = \sum_{i=i^*}^m \frac{\sigma_i}{1+e^{\beta\sigma_i}} \\
\leq& e^{-\beta\sigma_{i^*}} \sum_{i=i^*}^m \sigma_i
\end{aligned}$$

Therefore, $\zeta(t,\hat{t})$ can be upper bounded as below:

$$\zeta(t,\hat{t}) = \max_s \delta(s,t,\hat{t}) \leq \max_s e^{-\beta\sigma_{i^*}} \sum_{i=i^*}^m \sigma_i$$

By applying Claim 2, we have the following

$$\begin{aligned}
&Q^\pi(s,a,t,\hat{t}) - Q^\rho(s,a,t,\hat{t}) \\
\leq& \sum_{j=t+\hat{t}+1}^{T} \zeta(t,j-t) \\
\leq&(T-t-\hat{t}) \max_{t+\hat{t}+1 \leq j \leq T} \max_s e^{-\beta\sigma_{i^*}} \sum_{i=i^*}^m \sigma_i
\end{aligned}$$

Note that $\sigma_i \leq \max_{s,a,t,\hat{t}} Q^\pi(s,a,t,\hat{t}) \leq \sum_{t=0}^T d(t)R_{max}$, and $R_{max} = \max_{s,a} R(s,a) > 0$. If we choose $\sigma^* = \min_{i,s,t,\hat{t}} \sigma_i$ such that $\sigma_i > 0$ holds, the following bounds holds for all $(s,a,t,\hat{t})$ and $\beta > \beta_0$:

$$\begin{aligned}
&Q^\pi(s,a,t,\hat{t}) - Q^\rho(s,a,t,\hat{t}) \\
\leq&\big[m(T+1)\sum_{t=0}^T d(t)R_{max}\big]e^{-\beta\sigma^*}
\end{aligned}$$

□

C Discussion on the Action Nudge Formulation

We have formulated the design of action nudge as a linear program as in Equation (6). Note that there could be multiple actions that lead to $\phi(s,a,t) > 0$ for a given (s,t), implying that we might need to offer multiple competing nudge simultaneously. This might not be desirable in practice. One potential approach to address this is to add small random noises into $R^p(s,a)$. Since the reason for multiple competing nudges is due to ties (the agent is indifferent between choosing multiple actions), and the chance for two summations of random real numbers to be equal is small, this approach could reduce the the chance of having multiple nudges simultaneously. Below we can show that there exists an optimal solution such that this "multiple competing nudge" scenario will happen at most once.

Lemma A2. *There exists an optimal solution ϕ^* for problem 6, such that there is at most 1 state-time $(\hat{s},\hat{t})$ that we can find two actions a_1, a_2 such that $\phi^*(\hat{s},a_1,\hat{t}) > 0$ and $\phi^*(\hat{s},a_2,\hat{t}) > 0$.*

Proof. The problem formulated in (6) is a LP problem with $|S|(T+1)+1$ constraints, with $|S|T$ constraints on transition dynamics, $|S|$ constraints on initial distribution, and 1 constraint on the nudge budget (excluding the constraints $\phi(s,a,t) \geq 0$). Therefore, using the property of linear programs, there exists at least one optimal solution with at most $|S|(T+1)+1$ non-zero variables (the one with the smallest number of non-zero variables is called the basic feasible solution).

First consider a special case that we can find an optimal solution ϕ^* that (1) has at most $|S|(T+1)+1$ non-zero variables and (2) there exists an action a for every (s,t) such that $\phi^*(s,a,t) > 0$. Note that finding ϕ^* satisfying (1) is always possible using the property of linear programs as discussed above. If there exists ϕ^* that satisfies both conditions, the proof of the lemma is straightforward. Since we have $|S|(T+1)$ sets of (s,t), there will be at least $|S|(T+1)$ non-zero variables due to the condition. Since there is also at most $|S|(T+1)+1$ non-zero variables in ϕ^*, we can conclude there exists at most one set of (s,t) such that there contains two non-zero variables in ϕ^*.

Below we consider the general case that we can only find ϕ' that satisfies the first condition but not the second. We demonstrate how to construct a "smaller" problem that satisfies both conditions in a smaller problem instance. We then argue the optimal solutions in the smaller problem space is also optimal in the original space. First, we know there always exists a solution $\phi^{'}$ that satisfies the first condition from the property of linear programs. Now let us construct a "smaller" problem of the original problem (6). For a given $\phi^{'}$ that satisfies the first condition, denote $Y = \{(s,t)|\forall s,t\}$, the set of all (s,t), and $X = \{(s,t)|\sum_a \phi^{'}(s,a,t) = 0\}$, the set of (s,t) such that $\sum_a \phi^{'}(s,a,t) = 0$. Now construct an updated problem from problem (6) such that the the set of state-time pair is $Y \setminus X$ (i.e., the set of state-time pair that is in Y but not in X), but the transition and cost is still the same, and corresponding action probability is set to zero, i.e., $\phi(s,a,t) = 0$ if $(s',t+1) \in X$ and $P(s'|s,a) > 0$. Note that ϕ' is still the optimal solution in new problem, and any optimal solution in new problem is also going to be optimal in the original problem (since they perform at least as well as ϕ'). Note that now in the new problem, the number of constraint is $|S|(T+1)+1-|X|$, and the number of state-time pair is $|S|(T+1)-|X|$. Again, using the property of linear program, there exists a solution with at most $|S|(T+1)+1-|X|$ elements. If we can find a solution ϕ^* that satisfies the above while also satisfying the condition that there exists an action a for every $(s,t) \in Y \setminus X$ such that $\phi^*(s,a,t) > 0$, there exists at most one (s,t) with two nonzero $\phi^*(s,a,t)$ and we have the proof. If not, we can continue the above process to keep shrink the set of (s,t) until we find the new problem that satisfies both conditions. Note since in each new construction, the set of (s,t) is reduced at least by 1, and therefore this procedure will terminate in a finite number of times. This concludes our proof. □

In Lemma A2, we demonstrate the existence of solution such that "multiple competing nudge" only happens once. Now we show how to leverage the simplex method to find such a solution. The process is essentially an implementation of the procedure in the proof using the simplex method. Given problem (6), we can first use the simplex method to find a basic feasible solution, named ϕ^1. If ϕ^1 satisfies the requirement that "multiple competing nudge" happens at most once, then ϕ^1 is the desired solution. If not, since ϕ^1 is a basic feasible solution, there must exists some (s,t) such that $\phi^1(s,a,t) = 0, \forall a$. We could then reconstruct a problem by removing all those state-time in problem (6), then resolve the problem using simplex method to find a new basic feasible solution ϕ^2. Note that ϕ^2 has the same performance of ϕ^1 in the original problem. If ϕ^2 meets the requirement, we have found the desired solution. If not, we could repeat above process to construct new problem and find new solution, until the solution satisfies at most one "multiple nudge". Note that each new solution is still an optimal solution in original problem. In the worst case, we need to repeat the above process $|S|(T+1)$ times, i.e., the number of state-time pair. However, this is still linear in terms of time complexity. By following the above procedure, we can find a solution that "multiple competing nudge" happen at most once in polynomial time.

D Additional Simulation Results

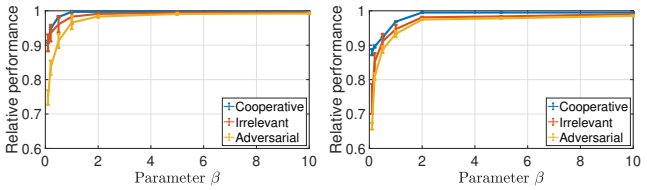
We present two additional sets of simulation results in this section. The first one examines the choice of β in the relaxed formulation in environment design via reward modification. The second one examines the scenario when the agent reward function and biases are not known a priori and evaluate whether we can leverage inverse reinforcement learning to infer agent rewards and biases to be used in our algorithms.

D.1 The effect of β in the relaxed formulation

We have shown that solving the environment design problem as defined in (3) is NP-hard and have proposed an relaxed formulation as in (4). The key parameter of this relaxation is β in the soft-max function. When $\beta \to \infty$, the relaxation is the

same as the original environment design problem. However, in practice, we can only solve it with a finite β.

Here we examine how much the choices of β impacts the outcome. We consider the mis-alignment of the principal's and the agent's reward function. For the agent model, we consider the boundedly-rational agent with $\tau = 1$ and present-bias agent with $k = 1$ (we have examined a range of different parameters, and the results are qualitatively similar.). For comparison, we brute-forcedly derive the optimal solution using bi-level solver of Pyomo[7] and examine how fast the performance of our algorithms converges to the true optimal as β increases. As shown in Figure A2, the performance converges quickly with β increases. It suggests that setting a small β is enough to reach reasonable approximations. This result also complements Lemma 3, proving the convergence of Q functions, and demonstrates that we can approximate the overall performance of the optimal.



(a) Bounded rational $\tau = 1$. (b) Present biased $k = 1$.

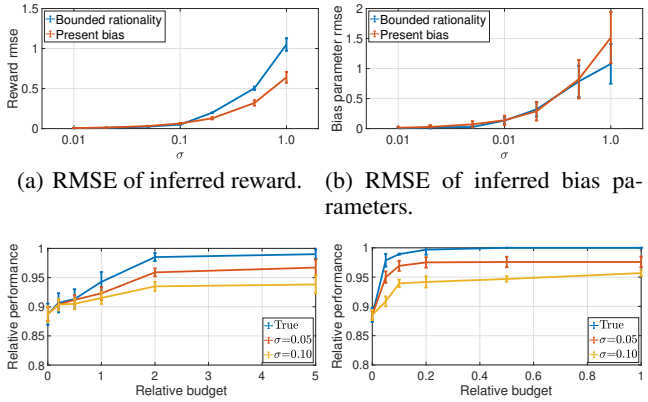
Figure A2: Examining the impact of β to the relaxation algorithm. When β is large, the relaxation is close to optimal. The results suggest that a small β is sufficient for the approximation.

We have also measured the runtime improvements for the relaxation. In our simulation ($|S|=100$, $|A|=4$, and $T=20$), it takes 7.9 seconds for our algorithm to solve an instance on average in our relaxed formulation while it takes 721.3 seconds to solve the instance exactly. The results demonstrate the efficiency improvements of the relaxation.

D.2 Unknown agent reward and biases: Combining inverse reinforcement learning

In our setting, we assume the reward function and agent bias parameters are known. While this assumption might be approximately satisfied in some cases (e.g., reward functions are payments specified by the system, and biases can be roughly estimated as in our experiment), it might not be satisfied in other cases. When these parameters are unknown, if we have access to data of human behavior in the original environment (e.g., user action history on the website), we might apply standard approaches in inverse reinforcement learning to simultaneously infer the reward and human biases first and then use the inferred values for environment design. In this set of simulations, we examine whether this idea is feasible.

We use the same simulation setup as in the main paper and apply the techniques by Evans and Goodman [2015] to infer the reward and bias parameter at the same time from the policy. Since they take a Bayesian approach, and the prior (initial belief about the parameters) would influence the outcome, we run simulations by assuming the prior is a noisy observation of the truth. In particular, let $r(s,a)$ be the true reward. In the prior, we randomly draw the prior of $r(s,a)$ to be $N(N(r(s,a),\sigma),\sigma)$, where $N(\mu,\sigma)$ is a normal distribution with mean μ and variance σ. Intuitively, larger σ implies a worse prior. We set the agent model to be a bounded rational agent with $\tau = 1$ (we have tried other agent models and the results are qualitatively similar). Figure 3(a) and 3(b) demonstrate how well the inverse reinforcement learning can estimate the true values with different noise σ in the prior. We then run our environment design algorithms on the inferred values, and Figure 3(c) and 3(d) demonstrate that our algorithms work on inferred rewards and biases as long as we have reasonable initial prior. While the results in this simulation are exploratory, it showcases the possibility to utilize environment design even when the rewards and human biases are initially unknown.



(a) RMSE of inferred reward. (b) RMSE of inferred bias parameters.

(c) Reward modification. (d) Action nudge.

Figure A3: Performance of environment design when the rewards and biases parameters are inferred.

E Details of the Human Subject Experiment

We provide the more detailed description of our human subject experiments in this section. We also summarize the demographic information of the online workers recruited from Amazon Mechanical Turk in the human-subject experiments.

E.1 Experiment design and task interface

Our human-subject experiment is approved by the IRB board in our institution. In our human-subject study, each worker is asked to play six navigation games, representing the decision-making environments. Similar to our simulation setting, each navigation game is represented by a grid world of size 10×10. The initial state is in the middle of the grid world, and the time horizon T is set to 20. In order to reduce the cognitive burden for human subjects, the reward function is simplified to only depend on the state, and we let the principal's reward function to be equal to the agent's reward function, i.e., $R^a(s,a) = R^p(s,a) = R(s)$.

The reward on each state is an integer from 1 to 100. Similar to the setup in the simulation, we place a high reward state (uniformly drawn from 80 to 100) in a random corner of grid

[7] https://github.com/Pyomo/pyomo

as global optimal, and a medium reward state (from 50 to 80) in other three corners as local optimal. Since the initial point is in the middle of map, we set the reward of path to global optimal and one local optimal to be low (from 10 to 30), and reward towards other two local optimal to be relatively high (from 30 to 50). Other places of map is set to be relatively low (from 1 to 30).

The interface of the navigation game is shown in Figure A4. Workers can move the plane around the map to collect the points in the grid world, and their bonuses depend on the total points they collected for the six games. For every 100 points collected, they can earn an additional USD $0.01 bonus. Taking into account the workers' working time in the task, and the $0.50 base payment for submitting the task, the average hourly rate is around USD $11.50.

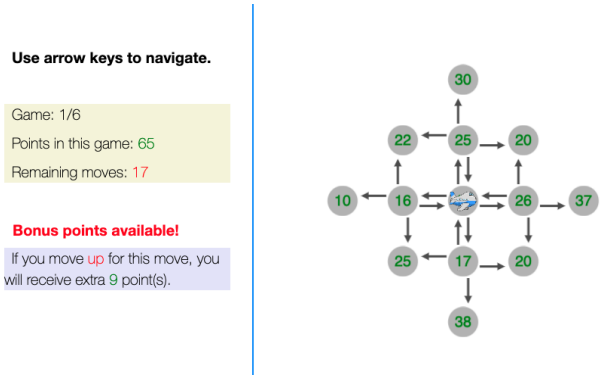


Figure A4: The interface of the navigation game in the human-subject experiment. Workers can use arrow keys to move the airplane around and collect points. The information on the bottom of the left-panel is the action nudge presented to workers, which is shown in action nudge treatment hen a nudge is provided by the principal, hidden otherwise.

To induce biased human behavior, at each time step, a worker can only see the rewards of the nearby states (to simulate the short-sightedness). Out of six games, there are two games each for vision length of 1,2, 3, which we use short-sighted agent with $\tau = 0, 1, 2$ to model when solving the environment design problem. Note that the purpose of this design is to provide us an estimate of human biases to be used in environment design. Worker behavior might not follow the behavior model.

Each worker is randomly assigned to one of the three treatments: {baseline, modified reward, action nudged}, with 106, 86, 108 workers assigned to each. The games are drawn from the same pool for each treatment. In the baseline treatment, workers will play the drawn games without modifications. In the modified reward treatment, workers will see the modified rewards generated by our algorithm, while in the action nudge treatment, when the nudge happens, the workers will see an additional message indicating they might gain bonus for moving towards a certain direction (as shown in Figure A4).

E.2 Demographic Information of the Workers

We summarize the demographic information of our 300 online workers recruited from Amazon Mechanical Turk in the human-subject experiments in Table A1.

Group	Category	Number
Age	19 or younger	6
	20 to 29	112
	30 to 39	110
	40 to 49	35
	50 to 59	26
	60 or older	11
Gender	Female	121
	Male	174
	Other	5
Race / Ethnicity	Caucasian	196
	Black or African-American	18
	American Indian/Alaskan Native	7
	Asian or Asian-American	63
	Spanish/Hispanic	7
	Other	9
Education	High school degree	20
	Some college credit, no degree	19
	Associate's degree	13
	Bachelor's degree	216
	Graduate's degree	29
	Other	3

Table A1: Demographic information of the 300 participants from Amazon Mechanical Turk in our experiment.