Modular Reinforcement Learning with Discounting

Ruohan Zhang[1*], **Shun Zhang**[2], **Matthew H. Tong**[3†], **Constantin A. Rothkopf**[4], **Mary M. Hayhoe**[3], **Dana H. Ballard**[1]

***For correspondence:** zharu@utexas.edu (RZ)

Present address: †IBM Watson, Austin, TX, USA

[1]Department of Computer Science, The University of Texas at Austin, Austin, TX, USA, ; [2]Computer Science and Engineering, University of Michigan, Ann Arbor, MI, USA, ; [3]Center for Perceptual Systems, The University of Texas at Austin, Austin, TX, USA; [4]Cognitive Science Center and Institute of Psychology, Technical University Darmstadt, Darmstadt, Germany

Abstract Please provide an abstract of no more than 150 words.
The reinforcement learning based on Markov decision processes can be used to model natural human behaviors. A branch called inverse reinforcement learning aims to estimate the underlying reward function from observed behaviors. Given the richness of the natural environment and the limitations in cognitive resources, standard reinforcement learning might not be practical for modeling real-time human decisions. We propose modular reinforcement learning, a divide-and-conquer and deterministic approach, for modeling natural behaviors. A module is defined as a subtask of the global task including two variables: a reward and a discount factor. The latter, which is frequently overlooked, specifies the temporal or spatial discounting rate of a future reward. We formalize modular reinforcement learning where the utility function of a decision maker can be interpreted intuitively by a value surface. Based on our modular reinforcement learning and previous work, we develop a sparse modular inverse reinforcement learning algorithm that is more data efficient, and can estimate the discount factor in addition to the reward. The correctness and computational efficiency of modular inverse reinforcement learning was first evaluated by computer simulated navigation experiments. Then, human subjects performed a navigation task in a virtual environment designed to simulate natural behavior such as crossing an intersection, where subjects followed a path, intercepted targets, and avoided obstacles. We demonstrate that using modular reinforcement learning as a model, and modular inverse reinforcement learning to estimate the rewards and discount factors for each module, we can predict human navigation behaviors with high accuracy, across different subjects and under different task conditions. Long human navigation trajectories can be reproduced end-to-end by an artificial agent that is built and trained using our model. Thus we are able to estimate the underlying intrinsic reward value and discount factor of tasks such as obstacle avoidance in natural human behavior.

Introduction

Reinforcement is a cornerstone of biological and psychological models of animal behaviors. For a long time its study was exclusively based on experimental observations, but since breakthrough work by ***Sutton and Barto*** (***1998***), a rapidly increasing number of studies have used formal reinforcement frameworks to model human behavior. Studies have focused on large-scale human decision making (***Gershman et al., 2009***; ***Glimcher et al., 2007***) as well as neural underpinnings (***Doya et al., 2002***; ***Doya, 2008***; ***Glimcher, 2011***). Basic mechanisms of reinforcement learning, such as reward

estimation, temporal-difference error, and discount factors, have been linked to vast brain regions (***Haruno et al., 2004***; ***Holroyd and Coles, 2002***; ***Kawato and Samejima, 2007***; ***Lee et al., 2012***; ***Cardinal, 2006***; ***Doya, 2008***) through neuroimaging and lesion studies.

The primary focus of formal reinforcement learning (RL) has been on forward models that, given reward signals, can learn to produce policies, which specify action choices when immersed in an environmental state. However an important breakthrough of RL in behavior modeling is inverse reinforcement learning (IRL), which aims to estimate the underlying subjective reward, given behavioral data of a decision maker (***Ng and Russell, 2000***). IRL is an appealing method for modeling human behavior: A behavioral model can be quantitatively evaluated by comparing human behaviors with that of an artificial agent trained using the model with the estimated reward function.

Reinforcement learning for natural behaviors

There are many challenges in applying reinforcement learning models to human visually guided behavior. In a task such as crossing the road, a person must determine the direction of heading, avoid tripping over the curb, locate other pedestrians or vehicles and their direction of heading and so on. Each of these particular goals requires some evaluation of the state of the world in order to make an appropriate action choice in the moment. Thus in the context of normal behavior humans make continuous sequences of sensory-motor decisions to satisfy specific goals. This conceptualization of visually guided behavior as a sequence of sensory-motor decisions has been formalized within the framework of statistical decision theory (***Maloney and Zhang, 2010***; ***Wolpert and Landy, 2012***). Central to this approach is the evaluation of the reward of an action in bringing about a goal. Although the importance of the neural reward circuitry in understanding human sensory-motor behavior is well established (***Glimcher and Fehr, 2013***), there are many difficulties in applying RL models to human behavior, especially in natural environments, and so far experimental investigations have been almost exclusively restricted to simple laboratory paradigms. In addition, there are almost no formal attempts to model such natural behavior (***Hayhoe and Ballard, 2014***).

An important factor that makes RL difficult in modeling natural behavior is its sophistication and resulting computational burden as a general model for all reward-seeking behavior. In standard RL, the optimal behavior, or policy, can be obtained through either a model-based dynamic programming approach which requires the full knowledge of the environment when visiting the environment for the first time, or a model-free learning approach which requires repeated visit to the environment. However, both of these methods are unlikely to be the real-time decision-making strategy since the model-based approach puts a heavy burden on memory storage and computation, and model-free approach requires a large amount of experience, which both become infeasible since decision makers encounter new situations all the time due to the rich and ever-changing natural environments. So although it is mathematically sophisticated and more general, the standard RL may not be the appropriate human decision-making model for natural tasks. Instead, approximations based on standard RL must be made to explain natural behaviors. Two major approximations inspired by recent progress in studying human cognition are employed in our model: a divide-and-conquer strategy and a deterministic planning strategy.

One important approximation strategy is based on a divide-and-conquer, i.e., a *modular* approach (***Rothkopf and Ballard, 2013***; ***Samejima et al., 2003***; ***Sprague and Ballard, 2003***) to standard RL. The modular RL approach decomposes a task into *modules* where each module solves a subgoal of the original task, hence the approach is desirable for the original tasks that are computationally difficult to represent or solve efficiently. Generally an arbitrator is required to coordinate module policies and make global decisions.

Recent studies have explored the plausibility of a modular architecture for natural visually guided behavior. Modular decomposition for RL is inspired by various human experiments, which tend to support a hierarchical modular architecture (***Ballard et al., 2013***; ***Gershman et al., 2009***; ***Kawato and Samejima, 2007***; ***Tong et al., 2017b***). Humans have limited capacity for attention and

working memory hence a computational efficient cognitive model is appealing. Studies of a variety of natural behaviors suggest that complex tasks can be broken down into concurrent execution of modules, or microbehaviors (***Ballard et al., 2013***; ***Tong et al., 2017b***; ***Land et al., 1999***; ***Hayhoe et al., 2003***). Thus in the example of walking across the street, humans must select a particular sub-goal at a given moment and these sub-goals are executed sequentially. The plausibility of this conceptualization was recently examined by ***Tong et al.*** (***2017b***), and shown to be a good approximation to both gaze behavior and action choices where subjects walked along a path, avoiding obstacles and intercepting targets. Each of these particular behavioral sub-goals can be treated as an independent module. This leads to a view of the human brain as the centralized arbitrator mentioned above that divides and coordinates these modules. The current investigation explores the modular architecture for this behavior in more detail.

Additionally, a standard IRL algorithm faces difficulties in estimating the reward function for natural human behaviors, since it generally requires a large number of data samples which could be impractical for human experiments. Under the modular RL framework, a more sample-efficient IRL algorithm is possible (***Rothkopf and Ballard, 2013***).

Estimating the discount factor

A frequently overlooked behavioral variable in RL is the discount factor, that determines how much a decision-maker weighs future reward compared to the current reward. In the agent-environment interaction model, the standard RL typically treats the discount factor as a part of the environment and as fixed. An alternative approach is to view the discount factor as a subjective decision-making variable that is part of the agent and may vary. This view is supported by behavioral neuroscience studies claiming the magnitude of the discount factor is correlated with serotonin level in human subjects (***Schweighofer et al., 2008***). Decision-makers may exhibit between-subject variation as research in behavioral science has shown that individuals may have distinct discount factors (***Story et al., 2014***). Not only that, between-task differences may also exist, i.e., the same decision maker may use different discount factors for different tasks. Hence an extension for the previous modular RL models to include discount factors for different human subjects and tasks is necessary.

To illustrate the progress towards the above research goals we select navigation tasks as the testing environment. These have been canonical benchmark testing domains for standard RL/IRL algorithms, while they are also ideal for modular RL/IRL approaches since it is convenient to introduce multiple tasks in these environments. Throughout computer simulations are first conducted to validate the correctness of the proposed algorithm and demonstrate its advantages over standard approaches. Then virtual reality and motion tracking technologies are employed to collect human behavior data. These technologies allow safe and efficient collection of large amount of human behavioral data in a rich, natural and controlled setting.

Methods

Modular Reinforcement Learning

Reinforcement learning basics

A standard reinforcement learning model is formalized as a Markov decision process (MDP). The MDP models the interaction between an environment and a decision maker, which will be referred as an agent. Formally, an MDP is defined as a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ (***Sutton and Barto, 1998***), where:

- $\mathcal{S}$ is a finite set of environment states. Let s_t denote the agent's state at discrete time step t. The state encodes relevant information for an agent's decision.
- $\mathcal{A}$ is a finite set of available actions. Let a_t be the action agent chooses to take at time t. The agent interacts with the environment by taking an action in its observed state.

- $\mathcal{P}$ is the state transition function which specifies the probability $P(s'|s,a)$, i.e., the probability of entering state s' when agent takes action a in state s. The state transition function describes the dynamics of the environment that are influenced by an agent's action.
- $\mathcal{R}$ is a reward function. $R(s,a)$ denotes the scalar reward agent received of taking action a in state s.
- $\gamma \in [0,1)$ is a discount factor. The agent values future rewards less than immediate reward, therefore future rewards are discounted by parameter γ at every discrete time step. $\gamma = 0$ indicates that the agent is myopic and only seek to maximize the immediate reward.
- $\pi : \mathcal{S} \mapsto \mathcal{A}$ is called a policy of the agent, which specifies the probability of chosen each action in each state.

The purpose of a reinforcement learning agent is to find an optimal policy π^* that maximizes the longterm cumulative reward. Many of the RL algorithms are based on value function estimation. The action-value function (also called Q-value function) estimates an agent the expected longterm reward for taking an action in a given state, and follow policy π afterwards. Formally, the Q-value function conditioned on policy π is defined as (***Sutton and Barto, 1998***):

$$Q^{\pi}(s,a) = \mathbb{E}_{\pi}\{R_t | s_t = s, a_t = a\} = \mathbb{E}_{\pi}\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a\} \tag{1}$$

Modular Reinforcement Learning

The divide-and-conquer approximation of RL results in modular reinforcement learning, in which a **module** is a subtask of the original task. Each module is a simpler problem, so that its value function and policy can be learned or calculated efficiently. A module is also modeled by an MDP $\langle \mathcal{S}^{(n)}, \mathcal{A}, \mathcal{P}^{(n)}, \mathcal{R}^{(n)}, \gamma^{(n)} \rangle$, where n is the index of the nth module. Note that each module has its own state space, transition function, reward function, and discount factor, but the action space is shared between modules, since modules reside in a single agent.

Let N be the total number of modules. We use $Q^{(n)\pi^{(n)}}$ to denote module Q-value function of the nth module conditioned on module policy $\pi^{(n)}$. For simplicity, we will write $Q^{(n)}$ and drop $\pi^{(n)}$. Q without superscription denotes the global Q function (also drop global policy π). Modular RL algorithms sum module Q functions to obtain the global Q function (***Russell and Zimdars, 2003***; ***Sprague and Ballard, 2003***):

$$Q(s,a) = \sum_{n=1}^{N} Q^{(n)}(s^{(n)}, a) \tag{2}$$

In a natural environment, there can be multiple **module objects** of a module, e.g., several obstacles nearby to avoid. The number of objects of each module is denoted by $M^{(1)}, \ldots, M^{(N)}$. Note that for a certain module, its module objects share the same $Q^{(n)}$ since their module MDPs are identical. But at a given time they could be in different states, denoted $s^{(n,m)}$ for module n object m. To generalize the above equation:

$$Q(s,a) = \sum_{n=1}^{N} \sum_{m=1}^{M^{(n)}} Q^{(n)}(s^{(n,m)}, a) \tag{3}$$

The previous modular RL would stop at this point. A module action-value function $Q^{(n)}$ may be calculated from solving Bellman equations using dynamic programming or through standard learning algorithms with enough experience, which we argue to be infeasible for human performing natural tasks. $Q^{(n)}$ needs to be calculated efficiently with reasonable cognitive load. Applying the Bellman equation to $Q^{(n)}(s^{(n,m)}, a)$:

$$Q^{(n)}(s^{(n,m)}, a) = \sum_{s'^{(n,m)}} P(s'^{(n,m)} | s^{(n,m)}, a) \Big[R(s^{(n,m)}, a) + \gamma^{(n)} Q^{(n)}(s'^{(n,m)}, a') \Big] \tag{4}$$

Then, we introduce the deterministic planning approximation, namely that the agent would treat state transition as if it is deterministic:

$$Q^{(n)}(s^{(n,m)}, a) = R(s^{(n,m)}, a) + \gamma^{(n)} Q^{(n)}(s'^{(n,m)}, a') \tag{5}$$

Like the transition function, the reward function would be treated as if it is deterministic, and since each module only considers a single source of reward (a single module object):

$$Q^{(n)}(s^{(n,m)}, a) = \begin{cases} 0 + \gamma^{(n)} Q^{(n)}(s'^{(n,m)}, a') & s'^{(n,m)} \text{ is non-terminal} \\ r^{(n)} & s'^{(n,m)} \text{ is terminal} \end{cases} \quad (6)$$

where $r^{(n)}$ is the reward for the n the module. Combining these two cases, and assuming a policy that leads the agent directly to the module object, $Q^{(n)}(s^{(n,m)}, a)$ takes the following simple form:

$$Q^{(n)}(s^{(n,m)}, a) = r^{(n)}(\gamma^{(n)})^{d(s^{(n,m)}, a)} \quad (7)$$

where $\gamma^{(n)}$ is the discount factor, and $d(s^{(n,m)}, a)$ is the spatial or temporal distance between the agent and the module object m after taking action a at state $s^{(n,m)}$. Note ***Equation 7*** brings value function of RL back to the simplest form like the one in (***Doya, 2008***), by eliminating factors that are less likely to be considered by real-time decision makers. More discussion of this derivation can be found in ***Appendix 1***.

Visualizing modular reinforcement learning

Equation 7 bridges modular RL with an important planning method called artificial potential field (***Khatib, 1986***; ***Arkin, 1989***; ***Huang et al., 2006***). Similar to a potential field, we use a value surface to visualize the value function. Module objects have different influences on the surface. The reward controls the maximum height of the surface, and the discount factor controls temporal or spatial discounting rates. Value surfaces are computationally easy to compose, hence multi-module behaviors can be modeled by combining these surfaces, which leads naturally to the modular RL. The concept of value surfaces and their combination is illustrated in ***Figure 1***. Given a composed value surface as in ***Figure 1***f, an modular RL agent would choose actions that lead to a local minima in the value surface. A sequence of actions may look like the trajectory in ***Figure 2***a, which traverses through a sequence of local minimum.

Modular Inverse Reinforcement Learning

While reinforcement learning aims at finding the optimal policy given an MDP, inverse reinforcement learning (IRL) observes the agent behavioral policy in the form of state-action pairs (s_t, a_t), and attempts to infer the unknown reward function (***Ng and Russell, 2000***; ***Abbeel and Ng, 2004***; ***Ziebart et al., 2008***; ***Ramachandran and Amir, 2007***). Specifically, this work is largely based on the modular IRL algorithm in (***Rothkopf and Ballard, 2013***) which pioneered the first modular IRL algorithm.

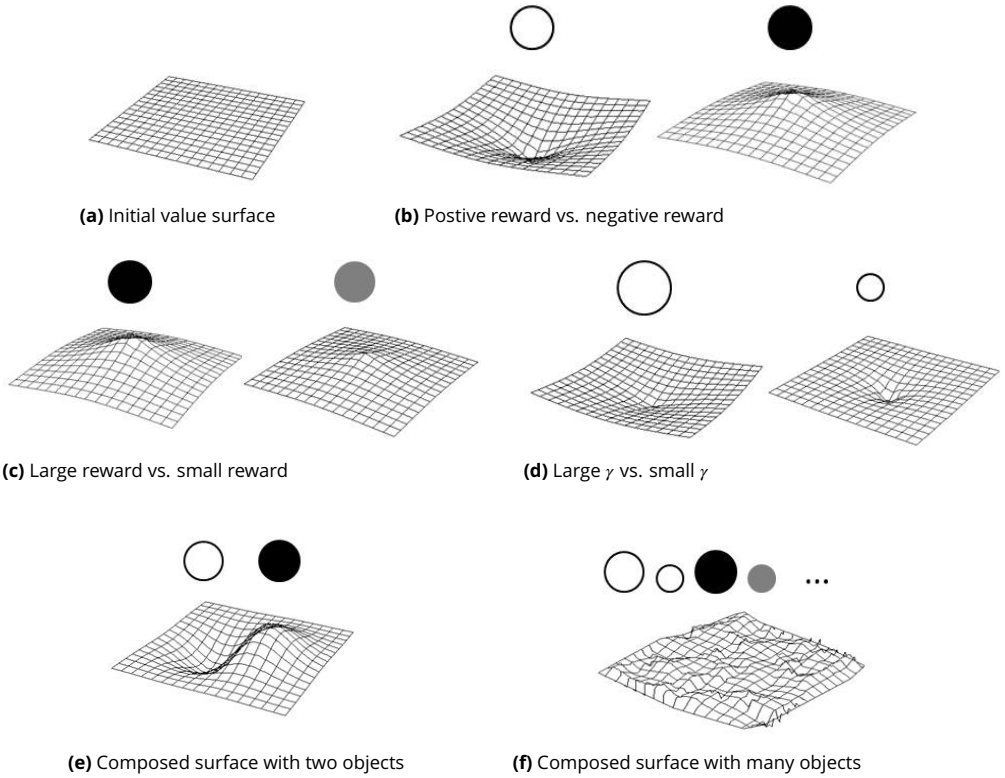
Given the modular RL formulation in the previous section, the goal of modular IRL is to estimate the underlying reward and discount factor for each module to recover the value function, given a sequence of observed state-action pairs (a trajectory) in state space as shown in ***Figure 2***a.

We follow the Bayesian formulation of IRL developed by (***Ramachandran and Amir, 2007***; ***Lopes et al., 2009***), Maximum Likelihood IRL in (***Babes et al., 2011***), and improve the modular IRL algorithm in (***Rothkopf and Ballard, 2013***). Let s_t denote the observed agent's state at time t, and a_t its chosen action. These approaches assume that the higher the Q-value for an action a_t in state s_t, the more likely action a_t is observed among all actions. Let η denote the confidence level in optimality (the extent to which an agent follows a greedy action selection, the default is 1), and let $\exp(.)$ denote the exponential function. The likelihood of observing a certain state-action pair is modeled by the softmax function with Gibbs (Boltzmann) distribution:

$$P(a_t | s_t, Q, \eta) = \frac{\exp(\eta Q(s_t, a_t))}{\sum_{a \in A} \exp(\eta Q(s_t, a))} \quad (8)$$

as illustrated in ***Figure 2***b.

Let T denote the total length of the trajectory. The overall likelihood $\mathcal{L}$ for observed data $D = \{(s_1, a_1), \cdots, (s_T, a_T)\}$ is the product of the likelihood of individual state-action pairs, given the



(a) Initial value surface **(b)** Postive reward vs. negative reward

(c) Large reward vs. small reward **(d)** Large γ vs. small γ

(e) Composed surface with two objects **(f)** Composed surface with many objects

Figure 1. The concept of modular reinforcement learning illustrated using value surfaces. (a) The initial value surface is flat without any reward signals. (b) A module object with positive reward has positive weight, and one with negative reward has negative weight. They bend the value surface to have negative and positive curvatures respectively. Therefore, an agent desires to follow the steepest descent to minimize the energy, or equivalently, to maximize the reward. (c) An object with larger weight bends the surface more. (d) An object with larger discount factor γ has greater influence over distance. (e,f) Composing different objects with different weights and γs, result in complicated value surfaces that seek to model agent's value function over the entire state space.

model is an MDP hence states are Markovian and action decisions are independent (***Rothkopf and Ballard, 2013***):

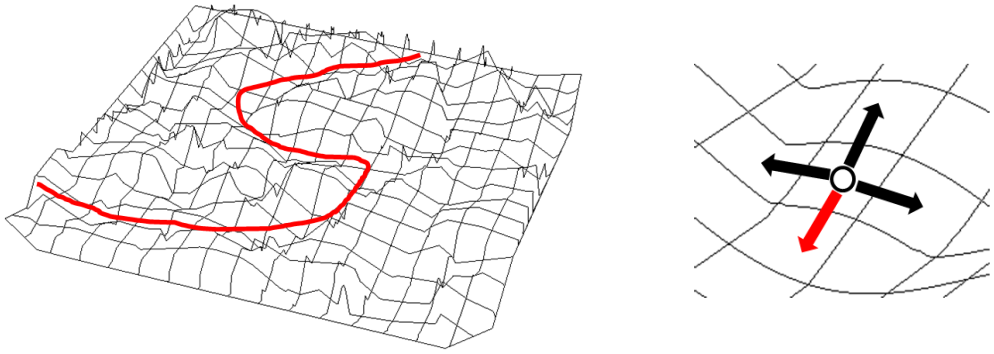
$$\mathcal{L} = P(D|Q,\eta) = \prod_{t=1}^{T} \frac{\exp(\eta Q(s_t,a_t))}{\sum_{a\in\mathcal{A}} \exp(\eta Q(s_t,a))} \tag{9}$$

Next, global $Q(s_t,a_t)$ is decomposed using ***Equation 3***. Given module Q functions $Q^{(1:N)}$, the likelihood becomes:

$$\begin{aligned}\mathcal{L} &= P(D|Q^{(1:N)},\eta) \\ &= \prod_{t=1}^{T} \frac{\prod_{n=1}^{N}\prod_{m=1}^{M_t^{(n)}} \exp(\eta Q^{(n)}(s_t^{(n,m)},a_t))}{\sum_{a\in\mathcal{A}}\prod_{n=1}^{N}\prod_{m=1}^{M_t^{(n)}} \exp(\eta Q^{(n)}(s_t^{(n,m)},a))}\end{aligned} \tag{10}$$

Take the log of the likelihood function:

$$\begin{aligned}\log\mathcal{L} = \sum_{t=1}^{T}\Big(&\sum_{n=1}^{N}\sum_{m=1}^{M_t^{(n)}} \eta Q^{(n)}(s_t^{(n,m)},a_t) \\ &- \log\sum_{a\in\mathcal{A}}\prod_{n=1}^{N}\prod_{m=1}^{M_t^{(n)}} \exp(\eta Q^{(n)}(s_t^{(n,m)},a))\Big)\end{aligned} \tag{11}$$



(a) (b)

Figure 2. Maximum likelihood modular inverse reinforcement learning. (a) From an observed trajectory (a sequence of state-action pairs), the goal of modular IRL is to recover the underlying value surface. (b) Maximum likelihood IRL assumes that the probability of observing a particular action (red) in a state is proportional to its Q value among all possible actions as in ***Equation 8***.

Substituting ***Equation 7*** into ***Equation 11***:

$$\log \mathcal{L} = \sum_{t=1}^{T} \Big(\sum_{n=1}^{N} \sum_{m=1}^{M_t^{(n)}} \eta r^{(n)} (\gamma^{(n)})^{d(s_t^{(n,m)}, a_t)} - \log \sum_{a \in \mathcal{A}} \prod_{n=1}^{N} \prod_{m=1}^{M_t^{(n)}} \exp(\eta r^{(n)} (\gamma^{(n)})^{d(s_t^{(n,m)}, a)}) \Big) \tag{12}$$

The variables to be estimated from data are the module rewards $r^{(1:N)}$ and discount factors $\gamma^{(1:N)}$. The number of modules N, the number of objects for each module $M_t^{(1)}, \ldots, M_t^{(N)}$, and distances $d(s_t^{(n,m)}, a_t)$ for each object are all state information and can be observed from the environment. This formulation follows closely the work by (***Rothkopf and Ballard, 2013***), extending it to use the new formulation of modular RL, handle multiple objects of each module, estimate the discount factors, and derive a slightly different objective function.

Sparse Modular Inverse Reinforcement Learning

Modular IRL can only guess which objects are actually in consideration by the decision maker. To alleviate this problem, we can further add a L_1 regularizer $-\lambda \sum_{n=1}^{N} ||r^{(n)}||_1$ to ***Equation 12***, which enforces some module rewards to be 0 so these modules would be ignored in decision making. This is an extension of the idea of using a Laplacian prior in Bayesian IRL (***Ramachandran and Amir, 2007***). In addition to the benefit from an optimization perspective, this term has the following important interpretation in terms of explaining natural behaviors.

A *hypothetical module set* is a set $\mathcal{H} = \{1, \cdots, N\}$ contains N modules that could potentially be of an agent's interest, e.g., all objects nearby. However, due to limitations in cognitive resource, the agent can only consider a subset of $\mathcal{H}$ at a time, denoted $\mathcal{H}'$. In a rich environment many modules' rewards would be effectively zero to the agent at current decision step, hence $|\mathcal{H}'| \ll |\mathcal{H}|$. For instance, a driving environment could contain hundreds of objects in $\mathcal{H}$. But a driver may pay attention to only a few. The regularization constant λ serves as a cognitive capacity factor that helps

determine $\mathcal{H}'$ from observed behaviors. Therefore, the final objective function of modular IRL is:

$$\max_{r^{(1:N)},\gamma^{(1:N)}} \sum_{t=1}^{T} \Big(\sum_{n=1}^{N} \sum_{m=1}^{M_t^{(n)}} \eta r^{(n)} (\gamma^{(n)})^{d(s_t^{(n,m)},a_t)} - \log \sum_{a\in\mathcal{A}} \prod_{n=1}^{N} \prod_{m=1}^{M_t^{(n)}} \exp(\eta r^{(n)} (\gamma^{(n)})^{d(s_t^{(n,m)},a)}) \Big) - \lambda \sum_{n=1}^{N} ||r^{(n)}||_1 \quad s.t.\ 0 \le \gamma^{(n)} < 1. \tag{13}$$

Another difference between our objective function and the one in (***Rothkopf and Ballard, 2013***) (Equation24) is that we do not normalize rewards in the objective. Since the main objective is derived from a softmax function, the rewards should be normalized after optimization.

Note that if we are to fit $r^{(1:N)}$ and $\gamma^{(1:N)}$ simultaneously, the above objective function is non-convex. However, the objective becomes convex if only fitting $r^{(1:N)}$. Since $\gamma^{(n)}$ is in range $[0, 1)$, one can perform a grid search over values for $\gamma^{(1:N)}$ with step size ϵ and fit $r^{(1:N)}$ at each possible $\gamma^{(1:N)}$ value. This allows us to find a solution within ϵ-precision of true global optima. Alternatively, various non-convex optimization techniques such as the differential evolution techniques (***Storn and Price, 1997***) can also be used.

A simulation-based evaluation of the modular IRL in a multitask navigation environment can be found in ***Appendix 2***, where the validity of the algorithm and its advantage over previous algorithms are shown.

Results

Despite the computational advantages of modular IRL as shown in simulation, the question remains to be whether it can be used as a decision-making model to explain actual human behaviors. In our experiments, virtual reality (VR) and motion tracking were employed to create a natural environment with rich stimulus, meanwhile to ensure experimental control. ***Figure 3*** shows the basic setup. The subjects wore a binocular head-mounted display (the nVisor SX111 by NVIS) that showed a virtual room. The subjects' eye, head, and body motion were tracked while walking through a virtual room. The subjects were asked to collect the targets (blue spheres) by intercepting them, follow the path (the gray line), and/or avoid the obstacles (red cubes). Thus the global task has three modules: following the path, collecting targets, and avoiding obstacles. Subjects were recruited from a subject pool of undergraduates at The University of Texas at Austin, and were naive to the nature of the experiment (***Tong et al., 2017b***). The human subject research is approved by The University of Texas at Austin Institutional Review Board approval number 2006-06-0085. The data is made public and available at (***Tong et al., 2017a***).

We gave subjects four types of instructions which attempt to manipulate their reward functions. This results in four experimental task conditions:

1. **Task 1**: Follow the path only
2. **Task 2**: Follow the path and avoid the obstacles
3. **Task 3**: Follow the path and collect the targets
4. **Task 4**: Follow, avoid, and collect together

Subjects received auditory feedback when colliding with obstacles or targets, and objects disappeared after collision. When objects were task-relevant, this feedback was positive (a fanfare) or negative (a buzzer), while collisions to task-irrelevant objects resulted in a neutral sound (a soft bubble pop). For further experimental details, see (***Tong et al., 2017b***). This general paradigm of navigation with targets and obstacles has been used to evaluate modular RL and IRL algorithms (***Sprague et al., 2007***; ***Rothkopf and Ballard, 2013***) and to study human navigation and gaze behavior (***Rothkopf and Ballard, 2009***; ***Tong et al., 2017b***).

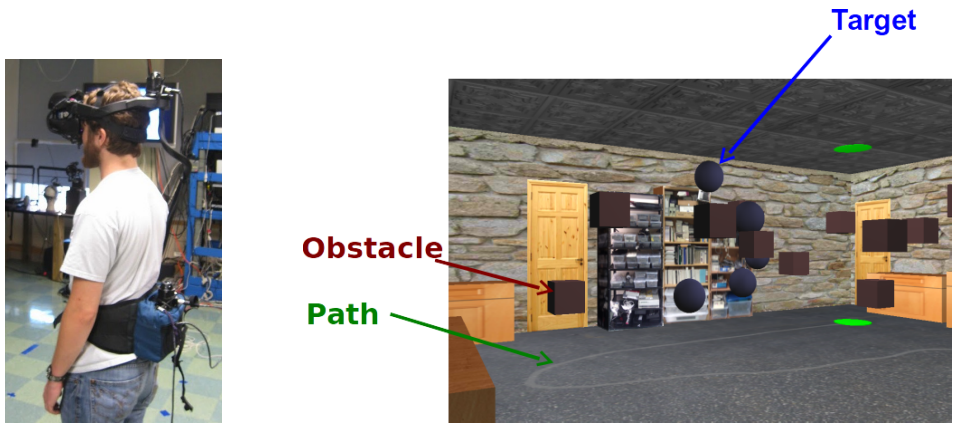


Figure 3. The virtual-reality human navigation experiment with motion tracking. (a) A human subject wears a head mounted display (HMD) and trackers for eyes, head, and body. (b) The virtual environment as seen through the HMD. The red cubes are obstacles and the blue spheres are targets. There is also a gray path on the ground leading to a goal (the green disk). At the green disk the subject is 'transported' to a new 'level' in a virtual elevator for another trial with a different arrangement of objects.

We analyze data collected from 25 human subjects. A single experimental trial consisted of a subject traversing the room, with the trial ending when the goal at the end of the path is reached. Objects positions and the path's shape differed on every trial. Each subject walked through the environment four times for each experimental condition. This results in a total of 100 experimental trials per experimental condition.

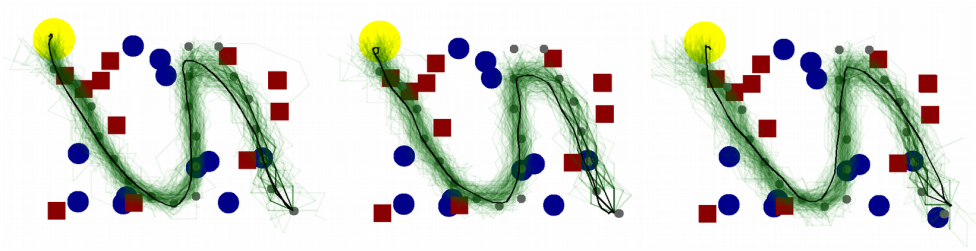
Sparse modular IRL ***Equation 13*** is used as the objective function to estimate r and γ. Different r and γ are estimated for different subjects under different task conditions, hence result in 25 subjects * 4 conditions = 100 different pairs of r, γ estimations. We use leave-1-out cross evaluation, where r, γ are estimated using all-but-one training trials that are from the same subject and same task condition and evaluated on the remaining test trial. The state information includes the distance and angle to the objects, while the state space is discretized using grids of size 0.572 by 0.572 meters, a parameter chosen via cross-validation that produces the best modeling result. The action space spans 360 degrees and is discretized to be 16 actions using bins of 22.5 degrees.

Qualitative results and visualization

An intuitive method to evaluate the modular RL model is to see whether the agent can accurately reproduce human navigation trajectories. The Q-value function for a modular RL agent can be calculated once r and γ are estimated. Next, a modular RL agent is placed at the same starting position as a human subject and starts to navigate the environment until it reaches the end of the path. The agent chooses an action probabilistically based on the Q-value of the current state, using a softmax action selection function with Gibbs distribution as in ***Equation 8***. A low value of parameter η makes the action selection uniformly random, we choose $\eta = 1$. The reason we let the agent choose the action with a certain degree of randomness is that in our environment, the Q-values for multiple actions can be very close, e.g., turning left or turning right to avoid an obstacle, and a human subject may choose either. Therefore, a single greedy trajectory may not overlap with the actual human trajectory. The softmax action selection function enables us to generate a distribution of hypothetical trajectories, i.e., a trajectory cloud, by running an agent many times in the same environment, and the human trajectory can be visualized in the context of this distribution.

Figure 4,***Figure 5***,***Figure 6*** and ***Figure 7*** shows selected trajectory clouds with the actual human paths, along with the corresponding rewards and discount factors. The agent trajectories are shown

in semi-transparent green hence darker area represents trajectories with higher likelihood, and actual human trajectories are shown in black. Each figure group presents experimental trials for one experimental condition (Task 1-4), and three trials within each row are from different subjects but the same environment, i.e., the same arrangement of objects.

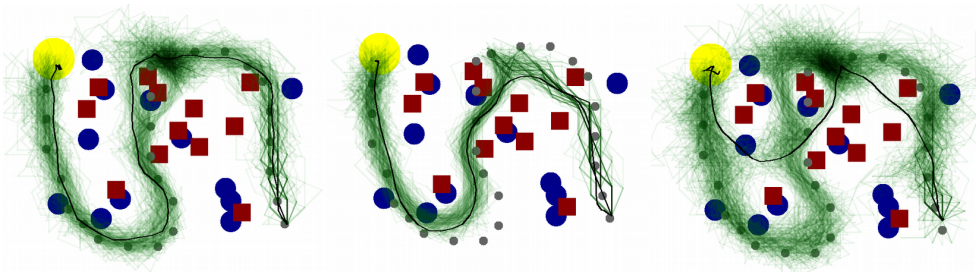


(a) r : $(0.11, 0.38, \mathbf{0.52})$
γ : $(0.85, 0.99, 0.88)$

(b) r : $(0.11, 0.25, \mathbf{0.63})$
γ : $(0.88, 0.99, 0.94)$

(c) r : $(0.07, 0.39, \mathbf{0.54})$
γ : $(0.82, 0.99, 0.94)$

Figure 4. Bird's-eye views of human trajectories and agent trajectory clouds across different subjects. Black lines: human trajectories. Green lines: modular RL agent trajectory clouds generated by softmax action selection. The green is semi-transparent hence darker area represents regions that are more likely to be visited. Yellow circles: destination points. Blue circles: targets. Red squares: obstacles. Gray dots: path. Below each graph are the rewards and discount factors estimated from human and used to train the modular RL agent. The rewards and discount factors are shown in the order of (Target, Obstacle, Path). The module rewards that correspond to task instructions are bold. Obstacle module has negative reward, but to compare with the other two modules the absolute value is taken. Three trials within each row are from different subjects but the same environment. This figure shows trials from **Task 1: follow the path**.



(a) r : $(0.00, \mathbf{0.57}, \mathbf{0.43})$
γ : $(0.00, 0.69, 0.91)$

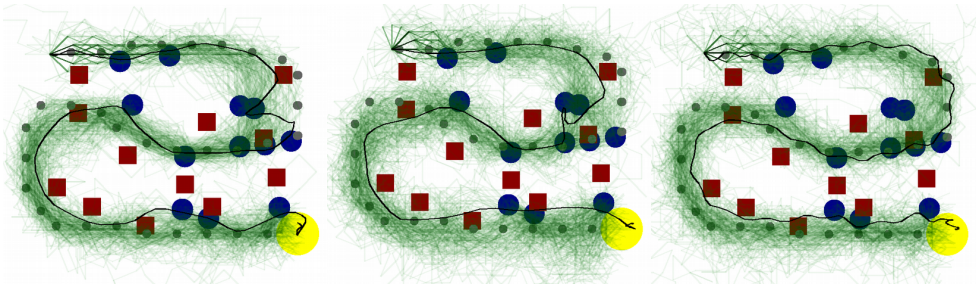
(b) r : $(0.02, \mathbf{0.57}, \mathbf{0.41})$
γ : $(0.99, 0.59, 0.97)$

(c) r : $(0.06, \mathbf{0.75}, \mathbf{0.19})$
γ : $(0.95, 0.60, 0.88)$

Figure 5. This figure shows trials from **Task 2: follow the path and avoid obstacles**. See ***Figure 4*** caption for details.

The main conclusion is that the model (generated trajectory clouds) align with data (human trajectories). Whenever a local trajectory distribution is multi-modal, e.g., in ***Figure 5***a, ***Figure 5***c, ***Figure 7***a, ***Figure 7***b, and ***Figure 7***c, the human trajectories align with one of the means. Note that the human trajectories do not always fall at the exact center of the distribution, this is expected because the rewards and discount factors are estimated not from the shown single trial, but using multiple trials that belong to the same subject and the same task conditions to obtain enough data samples.

The next important observation is the between-subject differences. Trials within each row are from the same environment under the same task instructions. However, human trajectories can sometimes exhibit drastically different choices, e.g., ***Figure 5***b versus ***Figure 5***c, ***Figure 7***a versus ***Figure 7***c. These differences are modeled by the underlying r and γ, and accurately reproduced by the distributions generated. This means that we can compactly model diverse human navigation behaviors using only a reward and a discount factor per module. The modeling power of modular



(a) r : $(\mathbf{0.30}, 0.22, \mathbf{0.48})$
γ : $(0.77, 0.72, 0.89)$

(b) r : $(\mathbf{0.27}, 0.29, \mathbf{0.45})$
γ : $(0.69, 0.73, 0.96)$

(c) r : $(\mathbf{0.30}, 0.17, \mathbf{0.52})$
γ : $(0.76, 0.99, 0.89)$

Figure 6. This figure shows trials from **Task 3: follow the path and collect targets**. See ***Figure 4*** caption for details.

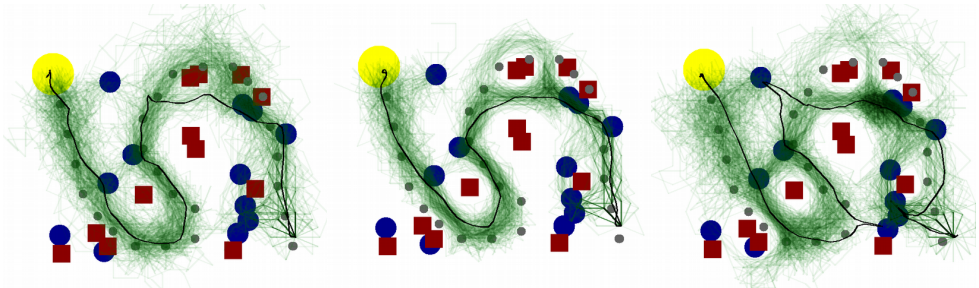
RL is demonstrated by the observation that varying these two variables can produce a rich class of navigation behaviors.

Between-task and between-subject differences

We then look at average reward and discount factor of all subjects, and compare them between tasks. The results are shown in ***Figure 8*** and ***Figure 9***. Overall, the estimated r agree with our task instructions, but things are not as simple as our binary relevant/irrelevant instructions would predict. For example although the reward values for collection and avoidance behavior are significantly higher when subjects are instructed to perform these tasks than when they are not, these modules are still given weight even in the irrelevant conditions. This effect is expected, since subjects most likely carry priors from previous experience; for example avoiding obstacles is presumably a highly learnt behavior. The model captures this by revealing the internal reward functions human subjects actually use. Furthermore, the between-subject differences in reward are shown in ***Figure 10*** for all 25 subjects. Again, changing in the relative reward between the modules is consistent with the task instructions. An one-way ANOVA test reveals that individual differences are evident across subjects under the same task instruction. See ***Appendix 3*** for a detailed analysis.

Quantitative results and ablation study

A quantitative evaluation metric would be the angular difference, i.e., the policy agreement, which is obtained by placing an agent in the same state as a human and measuring the angular difference between the agent action and the human action. Using this metric we also conduct an ablation study to demonstrate the relative importance of the variables in the modular RL model. The



(a) r : $(\mathbf{0.28}, \mathbf{0.34}, \mathbf{0.39})$
γ : $(0.72, 0.76, 0.90)$

(b) r : $(\mathbf{0.12}, \mathbf{0.60}, \mathbf{0.28})$
γ : $(0.74, 0.56, 0.97)$

(c) r : $(\mathbf{0.12}, \mathbf{0.77}, \mathbf{0.11})$
γ : $(0.74, 0.53, 0.95)$

Figure 7. This figure shows trials from **Task 4: follow the path, collect targets, and avoid obstacles**. See ***Figure 4*** caption for details.

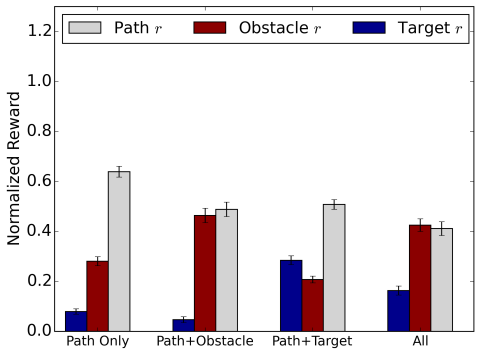


Figure 8. Normalized average rewards across different task instructions. The error bar represents the standard error between subjects. Obstacle module has negative reward, but to compare with the other two modules its absolute value is taken. In general the estimated reward agree with task instructions.

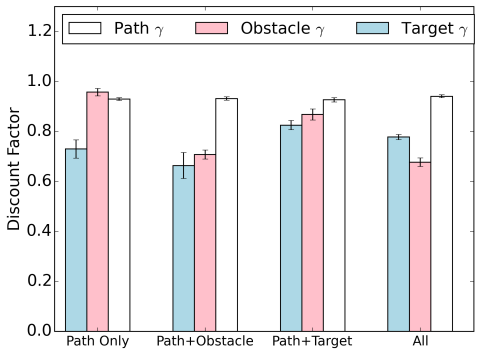
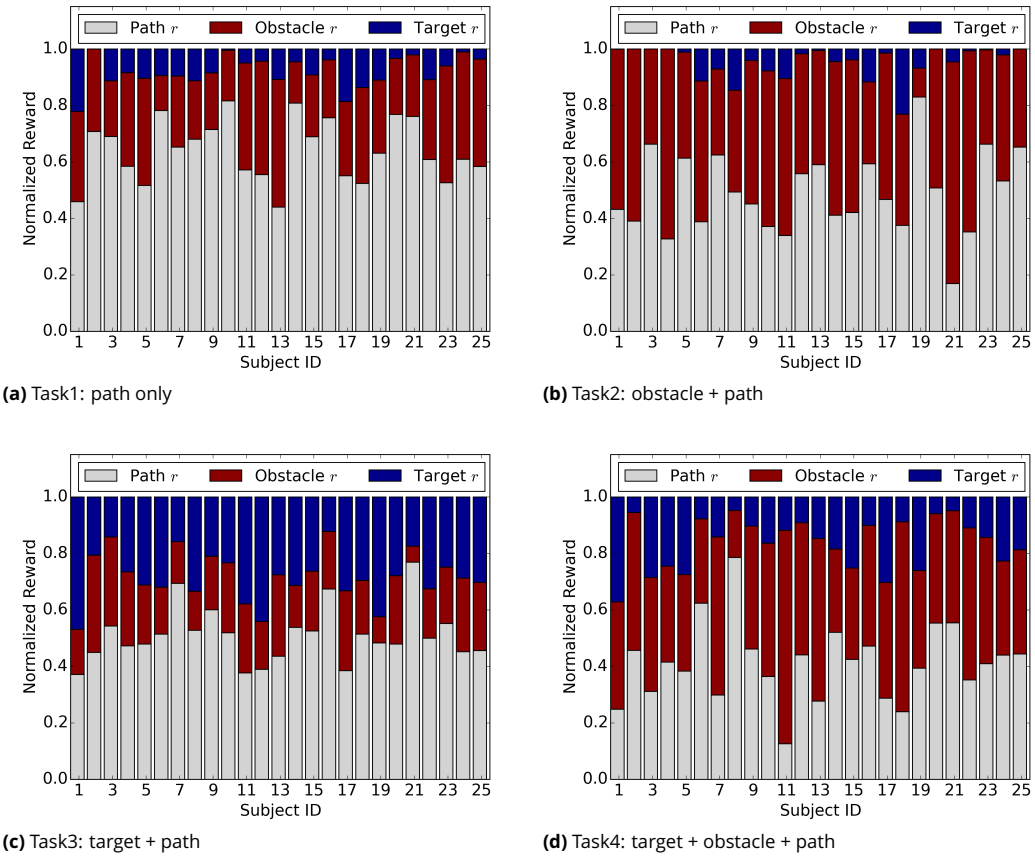


Figure 9. Average discount factors across different task instructions. The error bar represents the standard error between subjects.

comparison results are shown in ***Table 1***. The Random agent serves as a baseline which chooses an action uniformly random. The modular IRL full model agent chooses the greedy action that maximizes the Q-value function of each state, using both estimated r and γ. The binary reward agent only estimates γ, and use a reward unit of 1 for the module that is task-relevant, e.g., in Task 2 the path module and the obstacle module would have reward of +1 and -1 respectively, and the target module would have a reward of 0. The fixed γ agents estimate r only, and use $\gamma = 0.1, 0.5, 0.99$. The complete modular IRL agent is more accurate than other agents in predicting human actions, as shown by smaller angular differences in estimation.

The main conclusion from all human experiments and analyses above is that the modular RL agents generate reasonable hypotheses about underlying human decision-making mechanism. These results provides a strong support for using modular RL as the model for explaining such multitask navigation behaviors, and modular IRL as the algorithm to estimate rewards and discount factors. Such a model captures individual decision-making differences compactly with two behavioral variables per module while maintaining the modeling power. An ablation study shows that both reward and discount factors are significant. Another potential application of the model is to use the estimated reward to quantify the effects of experimental manipulation, especially when the experimental behavior is likely to be influenced by the priors in natural behaviors.



(a) Task1: path only

(b) Task2: obstacle + path

(c) Task3: target + path

(d) Task4: target + obstacle + path

Figure 10. Average rewards per subject under different task instructions. The relative reward magnitude changes between tasks. In addition, under the same task instruction, individual differences in reward function are shown. One-way ANOVA is performed for between-subject comparison.

Discussion

This paper formalizes a modular reinforcement learning model for natural multitask behaviors. The two important variables are the reward and discount factor, which can be estimated jointly from behavior data using the proposed modular IRL algorithm. A computer simulation demonstrated the sample efficiency of the modular IRL algorithm comparing to standard IRL. In a virtual-reality human navigation experiment, we showed multitask human navigation behaviors, across subjects and under different instructions, can be modeled and reproduced using modular RL. The modular RL/IRL not only yields computational advantages but also allows intuitive interpretations for multitask behaviors, where relative importance and reward discounting rates can be compared between subtasks directly.

An important strength of modular IRL is the ability to estimate the actual subjective value of behavioral goals. Over the last 15 years it has become clear that the brain's internal reward circuitry can provide a mechanism for the role of tasks on both gaze behavior and action choices. It is thought that the ventromedial prefrontal cortex and basal ganglia circuits encode the subjective values driving behavior (***Levy and Glimcher, 2012***; ***Bogacz et al., 2016***; ***Zénon et al., 2016***). Many of the reward effects observed for neurons with very simple choice response paradigms. Thus it is important to make the definitive link between the primary rewards used in experimental paradigms and the secondary rewards that operate in natural behavior. It seems likely that subjects learn stable values for the costs of particular actions like walking and obstacle avoidance and these subjective values factor into momentary action decisions, illustrated by small within-subject

Table 1. Evaluation of the modular agent's performance compared with baseline agents, measured by the angular difference (in degrees) compared to actual human decisions. The results are presented as mean ± standard error. The modular IRL agent outperforms all other models.

	Task 1	Task 2	Task 3	Task 4
Random	90.15±0.66	89.03±0.59	89.49±0.61	90.72±0.59
fixed γ = 0.1	31.74±0.88	39.43±1.18	36.16±0.75	41.40±0.88
fixed γ = 0.5	21.46±0.46	36.04±1.16	34.20±0.78	39.14±0.92
fixed γ = 0.99	18.19±0.32	27.63±1.41	28.61±0.93	31.63±1.08
Binary Reward	17.66±0.38	27.66±1.44	29.97±0.72	29.80±0.95
Modular IRL	17.94±0.33	27.39±1.46	26.98±0.80	27.65±1.02

variance of rewards. But relative values between tasks for different subjects vary, shown by the large between-subject variance of rewards.

Related work

The modular RL proposed in this work is most similar to a recent work in (***van Seijen et al., 2017***), in which they decompose the reward function in the same way as the modular reinforcement learning. Their focus is not on modeling human behavior, but rather on using deep reinforcement learning to learn a separate value function for each subtask and combining them to obtain a good policy. The modular RL proposed here bridges reinforcement learning with the artificial potential field methodology (***Khatib, 1986***; ***Arkin, 1989***; ***Huang et al., 2006***), a common planning method in robot navigation tasks. Modular RL can be explained and visualized as a potential field which corresponds to a value surface. Our modular IRL algorithm is an extension and refinement of (***Rothkopf and Ballard, 2013***) which introduced the first modular IRL and demonstrated its effectiveness using an simulated avatar. Our navigation task is similar to theirs, but we use data from actual human subjects.

The RL community has also developed other approaches to allow more efficient computation. A famous one is hierarchical RL (***Dietterich, 2000***; ***Sutton et al., 1999***) through *temporal abstraction*. In contrast with that approach, modular RL assumes *parallel decomposition* of the task. These two approaches are complementary, and are both important for understanding and reproducing natural behaviors. For example, a hierarchical RL agent could have multiple *options* (***Dietterich, 2000***; ***Sutton et al., 1999***) executing at a given time. A similar approach to modular RL is factored MDPs (***Guestrin et al., 2003***). The difference is that a factored MDP is a top-down, decomposition approach, in which the global MDP must be known beforehand. In contrast, modular RL is a bottom-up, ensemble approach. Another closely related approach is co-articulation (***Rohanimanesh and Mahadevan, 2005***), where a precedence relationship between components is used to determine a global policy.

Readdressing modular RL versus standard RL

In natural real-time navigation tasks like the experiments above, humans need to make decisions fast. Hence it might be difficult for them to calculate the actual global optimal policy using a standard RL algorithm. It is also unlikely for an agent to pre-compute the policy for all future states and use dynamic programming to obtain a global policy when they visit the environment for the first time as in our experiments. Doing so would at least require a human to store Q values for relevant states (a Q table) in its memory system, which is convenient for an artificial agent but would be difficult for a real-time human decision maker. Let's revisit two approximations made by modular RL:

- Divide and conquer: Modular RL assumes independent (transition, reward, and policy) modules in decision making, instead of a global planning strategy.

- Deterministic planning: Modular RL's action-value function for different states can be calculated without learning or storing the environment model, by treating transition and reward functions to be as if they are deterministic.

We argue that such a model makes it possible for the agent to calculate the action-value for a state when needed instead of beforehand, and demonstrated its performance experimentally. Additionally, the long range sensing capabilities of vision allow the discount factor for distant rewards to be estimated beforehand.

Evidence for the divide-and-conquer strategy

The intuition for a modularized strategy comes from two conjectures: learning is incremental and attentional resource is limited. A complicated natural task is often divided in to subtasks when learning happens, hence a real-time decision-making rule is likely to be a combination of pre-learned subroutines. A subtask is attended when needed to save computational resource. We expect this modular approach of RL can be applied to many natural tasks. Research (***Ballard et al., 2013***) has shown that complex human behavior can be modeled as consisting of microbehaviors, so we expect many behaviors are in fact a mixture of many simple modules and can be modeled in this way. Note that classic feature-based IRL algorithms (***Abbeel and Ng, 2004***; ***Levine et al., 2010***; ***Syed et al., 2008***) also divides the representation of the task MDP into features.

Re-examining deterministic planning

Why do we assume deterministic planning in natural tasks? What if an environment has probabilistic transition and reward functions? The first intuition simply comes from the fact that the act of sensing and storing state information is cognitively expensive in natural environments, which are required for model-based planning. Sensing the complete state information is expensive due to the foveated vision of human and the complexity of the environment. Storing all states information and planning a global optimal policy would also be difficult. Therefore, we argue that a human will plan when needed by treating the environment to be deterministic at current state, then observe and re-plan when needed if the environment proves uncertain. The second intuition comes from the research of (***Daw et al., 2011***; ***Skatova et al., 2013***), which have shown that learning in a MDP with even two stages requires significant amount of trials, given probabilistic transition functions. This indicates that planning in a probabilistic environment would require experience and learning. However, even with enough statistics to learn the model, human subjects could still prefer a deterministic learning (***Daw et al., 2011***; ***Skatova et al., 2013***).

Caveats and future work

Although the modular IRL is able to produce trajectories that are similar to human behavior, the match was imperfect as demonstrated by the angular difference. One difficulty with modeling human behavior is that we defined a state space and set of modules by hand without knowing the actual state representation or task decomposition that the human uses. This may account for the discrepancy between the human and agent policies. Ideally, we could learn state representation from data, but this involves the challenging task of combining representation learning and IRL. The work in ***Baker et al.*** (***2007***) provides a potential method for inferencing goals and states for the modules. Recent development in deep reinforcement learning (***Mnih et al., 2015***) may possibly lead to a data-driven approach to IRL that can learn representations.

In future work, we would examine an important assumption about the centralized arbitrator of the modules: agents form global Q-values by summing up module Q-values (***Russell and Zimdars, 2003***; ***Sprague and Ballard, 2003***). There has been work examining more sophisticated mechanisms for global decision making (***Bhat et al., 2006***; ***Ring and Schaul, 2011***). For example, one could schedule modules according to an attention mechanism instead of summing up module Q-values (***Bhat et al., 2006***; ***Zhang et al., 2015***). Whether these mechanisms can better explain human behaviors remains an open question that should be explored.

Additionally, the deterministic planning assumption could be further examined in a more controlled setting. When a human subject can visit a natural environment for many times, could model learning happen and decision strategy moves towards a model-based planning? How much motivation needs to be provided to elicit a more expensive model-based planning strategy? Will humans subjects exhibit clear individual differences in deterministic planning in natural tasks? These questions remain to be answered in future work.

Acknowledgments

References

Abbeel P, Ng AY. Apprenticeship learning via inverse reinforcement learning. In: *Proceedings of the twenty-first international conference on Machine learning* ACM; 2004. p. 1.

Arkin RC. Motor schema—based mobile robot navigation. The International journal of robotics research. 1989; 8(4):92–112.

Babes M, Marivate V, Subramanian K, Littman ML. Apprenticeship learning about multiple intentions. In: *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*; 2011. p. 897–904.

Baker CL, Tenenbaum JB, Saxe RR. Goal inference as inverse planning. In: *Proceedings of the 29th annual meeting of the cognitive science society*; 2007. .

Ballard DH, Kit D, Rothkopf CA, Sullivan B. A hierarchical modular architecture for embodied cognition. Multisensory research. 2013; 26:177–204.

Bhat S, Isbell CL, Mateas M. On the difficulty of modular reinforcement learning for real-world partial programming. In: *Proceedings of the National Conference on Artificial Intelligence*, vol. 21 Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999; 2006. p. 318.

Bogacz R, Moraud EM, Abdi A, Magill PJ, Baufreton J. Properties of neurons in external globus pallidus can support optimal action selection. PLoS Comput Biol. 2016; 12(7):e1005004.

Cardinal RN. Neural systems implicated in delayed and probabilistic reinforcement. Neural Networks. 2006; 19(8):1277–1301.

Daw ND, Gershman SJ, Seymour B, Dayan P, Dolan RJ. Model-based influences on humans' choices and striatal prediction errors. Neuron. 2011; 69(6):1204–1215.

Dietterich TG. Hierarchical reinforcement learning with the MAXQ value function decomposition. J Artif Intell Res(JAIR). 2000; 13:227–303.

Doya K. Modulators of decision making. Nature neuroscience. 2008; 11(4):410–416.

Doya K, Samejima K, Katagiri Ki, Kawato M. Multiple model-based reinforcement learning. Neural computation. 2002; 14(6):1347–1369.

Gershman SJ, Pesaran B, Daw ND. Human reinforcement learning subdivides structured action spaces by learning effector-specific values. The Journal of Neuroscience. 2009; 29(43):13524–13531.

Glimcher PW. Understanding dopamine and reinforcement learning: the dopamine reward prediction error hypothesis. Proceedings of the National Academy of Sciences. 2011; 108(Supplement 3):15647–15654.

Glimcher PW, Fehr E. Neuroeconomics: Decision making and the brain. Academic Press; 2013.

Glimcher PW, Kable J, Louie K. Neuroeconomic studies of impulsivity: now or just as soon as possible? The American economic review. 2007; 97(2):142–147.

Guestrin C, Koller D, Parr R, Venkataraman S. Efficient solution algorithms for factored MDPs. Journal of Artificial Intelligence Research. 2003; p. 399–468.

Haruno M, Kuroda T, Doya K, Toyama K, Kimura M, Samejima K, Imamizu H, Kawato M. A neural correlate of reward-based behavioral learning in caudate nucleus: a functional magnetic resonance imaging study of a stochastic decision task. The Journal of Neuroscience. 2004; 24(7):1660–1665.

Hayhoe M, Ballard D. Modeling task control of eye movements. Current Biology. 2014; 24(13):R622–R628.

Hayhoe MM, Shrivastava A, Mruczek R, Pelz JB. Visual memory and motor planning in a natural task. Journal of vision. 2003; 3(1):6–6.

Holroyd CB, Coles MG. The neural basis of human error processing: reinforcement learning, dopamine, and the error-related negativity. Psychological review. 2002; 109(4):679.

Huang WH, Fajen BR, Fink JR, Warren WH. Visual navigation and obstacle avoidance using a steering potential function. Robotics and Autonomous Systems. 2006; 54(4):288–299.

Kawato M, Samejima K. Efficient reinforcement learning: computational theories, neuroscience and robotics. Current opinion in neurobiology. 2007; 17(2):205–212.

Khatib O. Real-time obstacle avoidance for manipulators and mobile robots. The international journal of robotics research. 1986; 5(1):90–98.

Land M, Mennie N, Rusted J. The roles of vision and eye movements in the control of activities of daily living. Perception. 1999; 28(11):1311–1328.

Lee D, Seo H, Jung MW. Neural basis of reinforcement learning and decision making. Annual review of neuroscience. 2012; 35:287.

Levine S, Popovic Z, Koltun V. Feature construction for inverse reinforcement learning. In: *Advances in Neural Information Processing Systems*; 2010. p. 1342–1350.

Levy DJ, Glimcher PW. The root of all value: a neural common currency for choice. Current opinion in neurobiology. 2012; 22(6):1027–1038.

Lopes M, Melo F, Montesano L. Active learning for reward estimation in inverse reinforcement learning. In: *Machine Learning and Knowledge Discovery in Databases* Springer; 2009.p. 31–46.

Maloney LT, Zhang H. Decision-theoretic models of visual perception and action. Vision research. 2010; 50(23):2362–2374.

Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, et al. Human-level control through deep reinforcement learning. Nature. 2015; 518(7540):529–533.

Ng AY, Harada D, Russell S. Policy invariance under reward transformations: Theory and application to reward shaping. In: *ICML*, vol. 99; 1999. p. 278–287.

Ng AY, Russell SJ. Algorithms for Inverse Reinforcement Learning. In: *Proceedings of the Seventeenth International Conference on Machine Learning* Morgan Kaufmann Publishers Inc.; 2000. p. 663–670.

Ramachandran D, Amir E. Bayesian inverse reinforcement learning. In: *Proceedings of the 20th International Joint Conference on Artificial Intelligence* Morgan Kaufmann Publishers Inc.; 2007. p. 2586–2591.

Ring M, Schaul T. Q-error as a selection mechanism in modular reinforcement-learning systems. In: *Proceedings of International Joint Conference on Artificial Intelligence*, vol. 22; 2011. p. 1452.

Rohanimanesh K, Mahadevan S. Coarticulation: An approach for generating concurrent plans in Markov decision processes. In: *Proceedings of the 22nd International Conference on Machine Learning* ACM; 2005. p. 720–727.

Rothkopf CA, Ballard DH. Image statistics at the point of gaze during human navigation. Visual neuroscience. 2009; 26(01):81–92.

Rothkopf CA, Ballard DH. Modular inverse reinforcement learning for visuomotor behavior. Biological cybernetics. 2013; 107(4):477–490.

Russell SJ, Zimdars A. Q-Decomposition for Reinforcement Learning Agents. In: *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*; 2003. p. 656–663.

Samejima K, Doya K, Kawato M. Inter-module credit assignment in modular reinforcement learning. Neural Networks. 2003; 16(7):985–994.

Schweighofer N, Bertin M, Shishida K, Okamoto Y, Tanaka SC, Yamawaki S, Doya K. Low-serotonin levels increase delayed reward discounting in humans. the Journal of Neuroscience. 2008; 28(17):4528–4532.

van Seijen H, Fatemi M, Romoff J, Laroche R, Barnes T, Tsang J. Hybrid Reward Architecture for Reinforcement Learning. arXiv preprint arXiv:170604208. 2017; .

Skatova A, Chan PA, Daw ND. Extraversion differentiates between model-based and model-free strategies in a reinforcement learning task. Frontiers in Human Neuroscience. 2013; 7:525.

Sprague N, Ballard D. Multiple-goal reinforcement learning with modular Sarsa (O). In: *Proceedings of the 18th international joint conference on Artificial intelligence* Morgan Kaufmann Publishers Inc.; 2003. p. 1445–1447.

Sprague N, Ballard D, Robinson A. Modeling embodied visual behaviors. ACM Transactions on Applied Perception (TAP). 2007; 4(2):11.

Storn R, Price K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization. 1997; 11(4):341–359.

Story GW, Vlaev I, Seymour B, Darzi A, Dolan RJ. Does temporal discounting explain unhealthy behavior? A systematic review and reinforcement learning perspective. Frontiers in behavioral neuroscience. 2014; 8.

Sutton RS, Barto AG. Introduction to reinforcement learning. MIT Press; 1998.

Sutton RS, Precup D, Singh S. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. Artificial intelligence. 1999; 112(1):181–211.

Syed U, Bowling M, Schapire RE. Apprenticeship learning using linear programming. In: *Proceedings of the 25th international Conference on Machine Learning* ACM; 2008. p. 1032–1039.

Tong MH, Hayhoe MM, Zohar O, Zhang R, Ballard DH, Zhang S, Multitask Human Navigation in VR with Motion Tracking; 2017. https://doi.org/10.5281/zenodo.255882, doi: 10.5281/zenodo.255882.

Tong MH, Zohar O, Hayhoe MM. Control of gaze while walking: task structure, reward, and uncertainty. Journal of Vision. 2017; .

Wolpert DM, Landy MS. Motor control is decision-making. Current opinion in neurobiology. 2012; 22(6):996–1003.

Zénon A, Duclos Y, Carron R, Witjas T, Baunez C, Régis J, Azulay JP, Brown P, Eusebio A. The human subthalamic nucleus encodes the subjective value of reward and the cost of effort during decision-making. Brain. 2016; 139(6):1830–1843.

Zhang R, Song Z, Ballard DH. Global Policy Construction in Modular Reinforcement Learning. In: *AAAI*; 2015. p. 4226–4227.

Ziebart BD, Maas A, Bagnell JA, Dey AK. Maximum entropy inverse reinforcement learning. In: *Proceedings of the 23rd national conference on Artificial intelligence-Volume 3* AAAI Press; 2008. p. 1433–1438.

Appendix 1

Algorithm Assumptions

Compared to a standard RL formulation, modular RL (***Equation 3***) and (***Equation 7***) has the following assumptions:

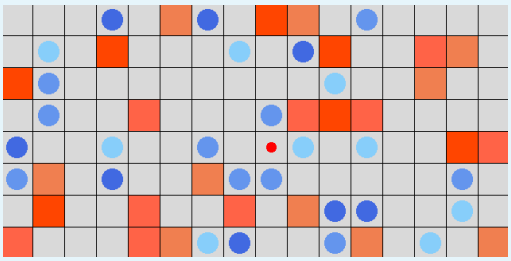
1. ***Equation 2*** and ***Equation 3*** assume that the global policy π associated with global Q function is a mixture of module policies. Thereby global Q function $Q^{\pi}(s,a)$ is the sum of $Q^{(n)\pi^{(n)}}(s^{(n,m)},a)$.
2. ***Equation 4*** assumes independent transition functions between modules, i.e., the existence of one module object does not affect another module object's transition function.
3. From ***Equation 4*** to ***Equation 6***, deterministic planning treats transition functions and reward functions as if they are deterministic, although they are determined by the dynamics of the environment.
4. From ***Equation 6*** to ***Equation 7*** there are two cases. First, if a reward is a positive reinforcement, ***Equation 7*** estimates the Q-value given the optimal policy that directly pursue the reward following a shortest trajectory in state space. Second, if a reward is a negative reinforcement, potential-based shaping reward (***Ng et al., 1999***) is added to each state to punish approaching that reward. The added shaping reward does not change the optimality of policy (***Ng et al., 1999***).

Assumptions 1 and 2 correspond to the divide-and-conquer strategy and assumption 3 corresponds to the deterministic planning. These assumptions distinguish modular RL from the standard RL, and the policy obtained from modular RL's Q-value function is generally not the optimal policy obtained from the standard RL. However, these mathematical assumptions are based on reasonable assumptions about human cognition that reduces computation loads. This important issue will be revisited later.

Appendix 2

Algorithm Validation

Using a canonical 2D gridworld in standard RL research, the goals are to validate that modular IRL algorithm can correctly estimate the rewards and discount factors, to demonstrate its advantage over the previous method, and to show an application of the sparse modular IRL. A portion of the gridworld is shown in ***Figure 1***. Different module objects are indicated by different colors and shapes. Behavior data (state-action pair samples) are collected from a modular RL agent.



Appendix 2 Figure 1. The 2D gridworld test domain. Red squares are obstacles with negative reward. Blue circles are targets with positive reward. The red dot is the modular RL agent. Different colors indicate different modules with distinct reward and discount factors. The objects of the same module are in the same color.

We first demonstrate that modular IRL is able to recover module rewards and discount factors correctly. The environment contains six modules each with ten objects. Three of them have positive rewards and the other three have negative rewards. 10 gridworlds are generated with random layouts of objects. The agent navigates each world for 6,000 steps. ***Equation 12*** is used to estimate $r^{(1:6)}$ and $\gamma^{(1:6)}$ and we calculate the mean estimation and standard deviation. The results are shown in ***Table 1***, it is evident that our algorithm is highly accurate in recovering rewards and discount factors given large amount of data as in this experiment.

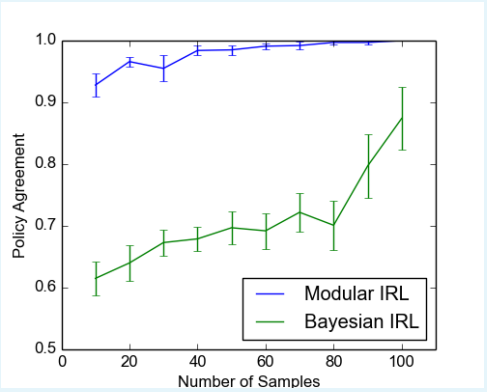
	$r^{(1)}$	$r^{(2)}$	$r^{(3)}$
Truth	+5	+10	+15
Estimation	+5.00±0.02	+9.94±0.03	+15.02±0.03
	$r^{(4)}$	$r^{(5)}$	$r^{(6)}$
Truth	-5	-10	-15
Estimation	-4.97±0.02	-10.03±0.03	-14.85±0.07
	$\gamma^{(1)}$	$\gamma^{(2)}$	$\gamma^{(3)}$
Truth	0.7	0.6	0.5
Estimation	0.70±0.00	0.60±0.00	0.50±0.00
	$\gamma^{(4)}$	$\gamma^{(5)}$	$\gamma^{(6)}$
Truth	0.3	0.2	0.1
Estimation	0.30±0.00	0.20±0.00	0.10±0.00

Appendix 2 Table 1. Estimated rewards and discount factors comparing to the ground truth for the six modules in the 2D gridworld experiment. The results are presented as mean ± standard error. The estimations are highly accurate due to the availability of a large amount of data.

Modular vs. standard IRL

In modeling natural behaviors, one particularly important aspect of an machine learning algorithm is its sample efficiency, given that it could be expensive to collect behavior data unlike in computer simulation. The performance of modular IRL on sample efficiency is

compared with a standard non-modular Bayesian IRL (***Ramachandran and Amir, 2007***). We use a Laplacian prior in Bayesian IRL since the rewards are sparse. ***Figure 2*** shows the results. The test environment has 4 modules and each has 4 objects. Both algorithms are tested with different number of samples (state-action pairs) and then compare the policies generated using the learned rewards. Policy agreement is defined as the proportion of the states that have the same policy as the ground truth, which is used because the outputs of these two algorithms are weights and rewards that can not be directly compared. Modular IRL obtained nearly 100% policy agreement with far fewer samples compared to the Bayesian IRL.

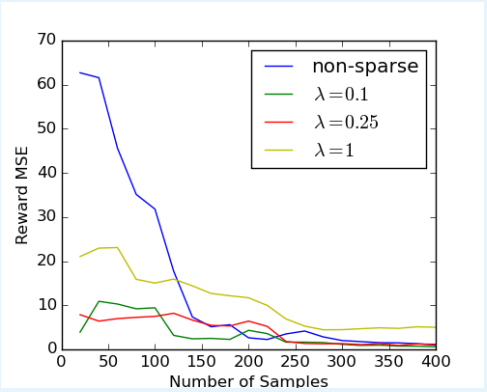


Appendix 2 Figure 2. Modular IRL vs Bayesian IRL on sample efficiency, measured by policy agreement. Modular IRL has significant higher sample efficiency.

Sparse modular IRL

Next we evaluate the performance of sparse modular IRL algorithm as in ***Equation 13*** on sample efficiency. The gridworld contains 10 modules and each has 10 objects. The agent has limited attention so it only considers 2 modules, i.e., the agent makes decision by treating all other modules to have zero rewards. Therefore, the hypothetical module set has size $|\mathcal{H}| = 10$ and actual module set $|\mathcal{H}'| = 2$. We use ***Equation 13*** to recover r and γ.

The mean squared error (MSE) of the estimated reward is shown in ***Figure 3***. If data is scarce, the sparse version of modular IRL algorithm ($\lambda = 0.1, 0.25$) can recover rewards more accurately than the non-sparse version. Sparse modular IRL correctly identifies modules that the agent paid attention to, indicated by low MSE values obtained. As the regularization constant λ controls the importance of the regularization term, a very large λ introduces too much bias in estimation and may fail to converge to the truth, as shown by $\lambda = 1$. One can use standard cross-validation techniques in choosing the value for λ.



Appendix 2 Figure 3. Modular IRL vs sparse modular IRL on sample efficiency, measured by mean squired error (MSE) of estimated reward. Sparsity can greatly improve sample efficiency with a carefully chosen value of λ.

Unlike computer simulated experiments where one can easily generate millions of samples, human experiments generally require more a expensive data collection process. Therefore the sample efficiency property of modular RL and IRL is an important advantage in modeling natural human behaviors.

Appendix 3

One-way ANOVA for Estimated Rewards and Discount factors

Appendix 3 Table 1. One-way ANOVA for individual differences in reward between subjects and across task instructions.

	Target r	Obstacle r	Path r
Task 1	$F(25,4) = 6.53$	$F(25,4) = 5.60$	$F(25,4) = 4.57$
	$p = 3.38\text{e}{-11}$	$p = 1.16\text{e}{-9}$	$p = 8.44\text{e}{-8}$
Task 2	$F(25,4) = 8.09$	$F(25,4) = 12.11$	$F(25,4) = 12.12$
	$p = 1.41\text{e}{-13}$	$p = 1.18\text{e}{-18}$	$p = 1.16\text{e}{-18}$
Task 3	$F(25,4) = 7.65$	$F(25,4) = 5.91$	$F(25,4) = 3.17$
	$p = 6.11\text{e}{-13}$	$p = 3.50\text{e}{-10}$	$p = 4.50\text{e}{-5}$
Task 4	$F(25,4) = 21.38$	$F(25,4) = 5.03$	$F(25,4) = 7.20$
	$p = 6.57\text{e}{-27}$	$p = 1.21\text{e}{-8}$	$p = 3.00\text{e}{-12}$

Appendix 3 Table 2. One-way ANOVA for individual differences in discount factor between subjects and across task instructions.

	Target γ	Obstacle γ	Path γ
Task 1	$F(25,4) = 1.99$	$F(25,4) = 1.37$	$F(25,4) = 17.96$
	$p = 0.01$	$p = 0.15$	$p = 2.94\text{e}{-24}$
Task 2	$F(25,4) = 2.10$	$F(25,4) = 4.17$	$F(25,4) = 14.20$
	$p = 6.72\text{e}{-3}$	$p = 4.77\text{e}{-7}$	$p = 7.63\text{e}{-21}$
Task 3	$F(25,4) = 2.23$	$F(25,4) = 5.73$	$F(25,4) = 10.91$
	$p = 3.65\text{e}{-3}$	$p = 7.16\text{e}{-10}$	$p = 2.91\text{e}{-17}$
Task 4	$F(25,4) = 0.60$	$F(25,4) = 4.93$	$F(25,4) = 10.65$
	$p = 0.92$	$p = 1.79\text{e}{-8}$	$p = 5.90\text{e}{-17}$

Appendix 4

Example Data

A sample video from collected human data. The videos shows a typical experimental trial from the subject's point of view, with motion tracking eye tracking enabled (the white cross). The task of this particular trial is to collect the targets, avoid the obstacles, and follow the path at the same time. A total of 100 trials are collected and used as the data for the modular inerse reinforcement learning to estimate the underlying reward functions and discount factors for these behaviors.