Combining imitation and deep reinforcement learning to human-level performance on a virtual foraging task

Journal Title
XX(X):1–12

DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Vittorio Giammarino[1], Matthew F Dunne[4,5,6], Kylie N Moore[4,5,6], Michael E Hasselmo[6], Chantal E Stern[4,5], Ioannis Ch. Paschalidis[1,2,3]

Abstract

We develop a framework to learn bio-inspired foraging policies using human data. We conduct an experiment where humans are virtually immersed in an open field foraging environment and are trained to collect the highest amount of rewards. A Markov Decision Process (MDP) framework is introduced to model the human decision dynamics. Then, Imitation Learning (IL) based on maximum likelihood estimation is used to train Neural Networks (NN) that map human decisions to observed states. The results show that passive imitation substantially underperforms humans. We further refine the human-inspired policies via Reinforcement Learning (RL) using the on-policy Proximal Policy Optimization (PPO) algorithm which shows better stability than other algorithms and can steadily improve the policies pre-trained with IL. We show that the combination of IL and RL match human performance and that the artificial agents trained with our approach can quickly adapt to reward distribution shift. We finally show that good performance and robustness to reward distribution shift strongly depend on combining allocentric information with an egocentric representation of the environment.

Keywords

Decision Making, Foraging, Reinforcement Learning, Imitation Learning, Autonomous Navigation, Deep Learning, Bio-inspired Control

Introduction

Human beings are exceptional learners: capable of conceiving solutions for individual problems, generalizing acquired skills to new tasks, exploring new strategies, and inferring causal relationships (Walker et al. 2012; Gopnik et al. 2015; Goddu et al. 2020; Ruggeri et al. 2021). Since the earliest stage of Machine Learning (ML), the research community has sought to emulate humans' learning capacities; and only recently, works which combine Deep Learning (DL) with Reinforcement Learning (RL), have accomplished outstanding results in this regard (Silver et al. 2017). DL and RL have shown to be indispensable ingredients to accomplish human-like intelligence in artificial systems; however, they still require a massive amount of computational resources and do not show the same level of efficiency compared to human beings (Botvinick et al. 2019). A viable option to tackle the efficiency issue, again inspired by human learning (Offerman and Sonnemans 1998; Jones 2009) is to leverage human demonstrations by combining DL and RL with imitation in a procedure known as imitation learning (IL) (Pomerleau 1991). It is worth noting that a great number of tasks, such as navigating or exploring unknown environments, are relatively straightforward for humans and can be successfully learned in a limited number of trials. On the other hand, this is often not the case for artificial agents, where the amount and the quality of the information retrieved, in addition to a sound design of the reward function and/or a good exploration

[1]Division of Systems Engineering, Boston University, Boston, MA 02446, USA.
[2]Dept. of Electrical and Computer Engineering, Boston University, Boston, MA 02446, USA.
[3]Dept. of Biomedical Engineering, Boston University, Boston, MA 02215, USA.
[4]Cognitive Neuroimaging Center, Boston University, Boston, MA 02215, USA.
[5]Graduate Program for Neuroscience, Boston University, Boston, MA 02215, USA.
[6]Center for Systems Neuroscience, Boston University, Boston, MA 02215, USA.

Corresponding author:
Vittorio Giammarino, Division of Systems Engineering, Boston University, Boston, MA 02446, USA.
Email: vgiammar@bu.edu

strategy of the environment, are crucial for successfully learning from scratch. Hence, artificial agents might benefit from directly imitating human behavior or, alternatively, from reconstructing human-inspired reward functions (Abbeel et al. 2010).

In this study, we investigate the potential of learning from humans taking into account not only performance but also data-efficiency, i.e. the amount of interactions with the environment an agent needs in order to master the task. We start by collecting movement data from a series of human participants while they are performing a virtual foraging task in which the rewards, in the form of coins, are condensed in clusters throughout the environment. The participants, subject to time constraints, have to collect the highest number of coins, effectively trading-off between foraging within a single cluster (exploitation) and exploration. Humans are initially unaware of the number of clusters and of their locations but are able to learn the reward distribution throughout the course of the experiment.

Note that, time-constrained foraging problems occur in several realistic scenarios, including scientific exploration; where a rover might want to sample chemical or geological features as fast as possible; search and rescue operations, where a vessel needs to rescue as many people as possible (Scone and Phillips 2010; Otte et al. 2013), wildlife tracking, agriculture pollination, agriculture harvesting and so on. Moreover, these missions can be dangerous and time demanding, and the use of aerial, marine or ground unmanned vehicles would significantly mitigate risks and save time. However, without assumptions about the distribution of targets, classical control techniques are not applicable. Teleoperation is also a feasible option, but it may be hindered due to unreliable communications, and this approach does not scale as well as completely autonomous options. For these reasons, ML techniques supporting full autonomy represent an interesting alternative solution. Therefore, our main objective is to develop a method which effectively combines IL with DL and RL and allows for efficient human-level learning in a foraging task with sparse rewards.

From our experiment, we collect 50 human trajectories and further process them to include both allocentric and egocentric information in our model. Where allocentric means the coordinates with respect to the environment frame and egocentric means the perception of the surrounding. We then run IL on each of the trajectories, yielding 50 policies with different performance. None of these policies succeed in matching human results. Finally, we use the imitated policies as initial solutions and further refine them with RL. By combining the two methods, we outperform the average human performance and the respective participant from which the agent imitates with a success rate of 78% and 62%, respectively, while using a reasonable amount of training steps ($\leq 10^7$). We also compare our method with a pure RL alternative, and show that such an approach remarkably underperforms humans.

In the final part of the work, we test the learned policies for generalization and robustness in a new scenario with an unknown reward distribution. We are able to show that the artificial agents quickly adapt to this new scenario and conjecture that, when combined with allocentric information, the egocentric representation of the environment plays a key role in enabling learning as also observed in neurophysiological recordings from rodents (Alexander et al. 2020). To empirically test this hypothesis, we rerun the entire set of experiments only considering allocentric coordinates. We show that in the absence of an egocentric representation of the environment, RL is unable to further improve IL and the final performance does not reach the level of human subjects. For the sake of completeness, we rerun the experiments also considering only the egocentric representation of the environment. This setup, however, violates the MDP assumption and, since our agents are not equipped with explicit memory, the learnt policies underperform those of the other experiments. A figure illustrating all these experiments for the full set of 50 human trajectories is included in the supplementary materials. We conclude that proper modeling is as crucial as the right algorithmic choices in order to enhance general and robust learning in artificial agents.

Related Work and Contribution

We focus on combining IL with RL in order to address the shortcomings of these two approaches when used individually. IL was initially proposed as a supervised learning method for faster policy learning (Pomerleau 1991; Schaal 1999). Recent works have studied the limitation of IL including the covariate shift problem and its dependency on the quality of the demonstrations (Syed and Schapire 2010; Ross and Bagnell 2010). RL instead was proposed to enable learning through direct interaction with the environment (Sutton and Barto 2018). RL combined with DL has achieved outstanding results in policy learning (Silver et al. 2017), however, sample inefficiency and safety remains an obstacle for its deployment in real world scenarios (Dulac-Arnold et al. 2019; Serrano-Cuevas et al. 2020). Recent works have endeavored to combine IL with RL either to address the limitations of IL (Ross et al. 2011; Ross and Bagnell 2014; Sun et al. 2018; Cheng et al. 2019) or to improve efficiency in RL (Kober et al.

2010; Subramanian et al. 2016; Vecerik et al. 2017; Nair et al. 2018; Libardi et al. 2021; Uchendu et al. 2021).

Another line of research, known as Inverse Reinforcement Learning (IRL), leverages demonstrations in order to infer a reward function which is then used for RL in order to recover the demonstrator policy (Abbeel and Ng 2004; Ratliff et al. 2006; Ziebart et al. 2008; Finn et al. 2016). IRL has the pros of being immune to the covariate shift problem but the cons of being as sample inefficient as the used RL algorithm. A unified view of IRL and IL as an f-divergence minimization problem has been recently proposed (Ho and Ermon 2016; Ghasemipour et al. 2020) and addressed using Generative Adversarial Networks (Goodfellow et al. 2020) (GAN) for either IL (Ho and Ermon 2016) or reward shaping (Kang et al. 2018).

We place our work in the Reinforcement Learning with Expert Demonstrations (RLED) framework, where the RL agent learns in the same environment of the demonstrator and using the same reward function (Hester et al. 2018). Our main contribution is leveraging a non trivial case study to show how modeling, imitation and reinforcement, when effectively combined, can lead to human-like performance in navigation tasks with sparse rewards without requiring a massive amount of training steps. Note that, this does not mean that RL-only algorithms cannot achieve human-level performance in this type of tasks, rather that they need a significantly larger number of steps to do so. Furthermore, dense reward signals would have most likely improved RL-only performance but also assumed prior knowledge of the environment invalidating therefore the comparison with the human agents. Finally, we analyze our results for robustness and empirically show a strong correlation between egocentric representation of the environment and performance. We reemphasize the importance of the right algorithmic choices as well as the right model in order to enhance effective learning in artificial agents.

Other works which likewise combine IL with RL are Hester et al. (2018); Rajeswaran et al. (2017); Uchendu et al. (2022) and Silver et al. (2017). However, AlphaGo in Silver et al. (2017) lies in the model-based spectrum, whereas, our work considers a pure model-free RL setting. Deep Q-Learning from Demonstrations (DQfD) in Hester et al. (2018), on the other hand, explores pre-training a deep Q network (Mnih et al. 2013) using demonstrations before performing RL. In order to do so, the agent needs access not only to the demonstrator state-action pairs but also to the rewards collected along the trajectories. In other words, DQfD assumes access to the full MDP transition (states, actions and rewards) and as a result, its pretrain step can be seen as a first form of offline RL (Levine et al. 2020). In our study, we assume access only to demonstrator state-action pairs (without rewards) and therefore DQfD is not directly applicable. Furthermore, none of the aforementioned works investigate the effects of modeling on algorithmic performance. Closer to our approach are the algorithms presented in Rajeswaran et al. (2017) and Uchendu et al. (2022). However, our study is focused on a unique foraging domain that poses particular challenges. In addition, we explore the resilience and adaptability of our artificial agents and draw comparisons with human performance.

Outline and Notation

The remainder of the paper is organized as follows. The Materials and Methods section presents the experimental setup used to collect the human foraging data, introduces the MDP model used for representing human behavior, discusses the IL for learning policies from data, and outlines the RL algorithms used to refine the imitated policies. In the Results section we compare our method with human and RL-only performance on the original setup; then, we test all the artificial agents for robustness to reward distribution shift and demonstrate the importance of egocentric information. We discuss the results in the Discussion section.

Unless otherwise indicated, we use uppercase letters (e.g., S_t) for random variables, lowercase letters (e.g., s_t) for values of random variables, script letters (e.g., $\mathcal{S}$) for sets, and bold lowercase letters (e.g., $\boldsymbol{\theta}$) for vectors. Let $[t_1 : t_2]$ be the set of integers t such that $t_1 \leq t \leq t_2$; we write S_t such that $t_1 \leq t \leq t_2$ as $S_{t_1:t_2}$. We denote by $\mathcal{N}(\mu, \sigma^2)$ the normal distribution, where μ is the mean and σ the standard deviation. $\mathbf{I}$ denotes the identity matrix. We denote the multivariate normal distribution with $\mathcal{N}(\boldsymbol{\mu}, \sigma^2\mathbf{I})$ where $\boldsymbol{\mu}$ is the mean vector and the covariance matrix is diagonal with only σ^2 as elements on the diagonal. Finally, $\mathbb{E}[\cdot]$ represents expectation and $\mathbb{P}(\cdot)$ probability.

Materials and Methods

Experimental Setup

In the following section, we provide a description of how the human foraging datasets were collected. In the supplementary materials we include the full dataset of 50 search trajectories. We focus on five participants in the context of a larger study investigating human foraging (Moore et al. 2021). An example of two foraging search trajectories is given in Fig. 2. All the experiments have been carried out in accordance with the relevant guidelines and regulations and approved by the Boston University's Institutional Review Board.

(a) (b) (c)

Figure 1. Fig. 1a and 1b are two snapshots of the foraging game in which the forager is about to collect coins. Fig. 1c shows a top view of the environment where the coins are indicated by crosses and the coin clusters are indicated by circles.

Participants consisted of male and female, neurologically healthy, English-speaking volunteers between the ages of 18-35 with normal or corrected-to-normal vision. Participants were recruited from Boston University and the surrounding community. Individuals with a history of drug abuse, use of psychoactive medication, neurological or psychiatric disorders, or learning disabilities were excluded. Additionally, participants with a history of motion sickness when watching or playing video games were also excluded. All participants were compensated and gave written informed consent in accordance with Boston University's Institutional Review Board.

The task consisted of a $160m \times 160m$ virtual "open-field", i.e. obstacle free, paradigm surrounded by four differently colored and textured walls created using Vizard 6.0, a Python-based virtual reality development platform (Fig. 1a). 325 coins were distributed throughout the environment, of which 100 were uniformly randomly distributed and 225 were distributed according to four different multivariate Gaussian distributions of varying sizes: 75 according to $\mathcal{N}((60, 75), 5^2\mathbf{I})$, 40 according to $\mathcal{N}((-15, -50), 11^2\mathbf{I})$, 60 according to $\mathcal{N}((-50, 30), 18^2\mathbf{I})$ and 50 according to $\mathcal{N}((49, -40), 13^2\mathbf{I})$ (Fig. 1c). Each participant's starting location was randomized at the beginning of each run. Participants could move forward and turn left or right. They could not move backwards. They were instructed to freely explore the environment and collect as many coins as possible but were not told anything about the distribution or total number of coins. They were also able to see a running count of the coins they had collected for each run. After being collected each coin disappears for the remaining duration of the run. Participants performed the foraging task over two consecutive days. On the first day, naive participants were presented with the task on a desktop computer in a behavioral testing room. On the second day, they performed the same task in an MRI scanner. Subjects performed 10 eight-minute runs on Day 1 and 10 eight-minute runs on Day 2. In the desktop condition (Day 1), participants moved using keyboard arrow keys, and in the scanner (Day 2), they moved using a diamond-shaped button box. For our purposes here, we utilize the 80 minutes of behavioral data from Day 2 of the experiment for 5 participants which collected an average of 243.98 coins each. Selecting behavioral data from Day 2 rather than Day 1 is motivated by the fact that, during Day 2, the participants were already familiar with the task and achieved improved performance compared to Day 1.

Modeling the Human Decision Process

In this section we describe the human modeling step and how the data are processed to make them suitable for IL and RL.

We consider an infinite-horizon discounted Markov Decision Process (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, P, r, D, \gamma)$ where $\mathcal{S}$ is the finite set of states and $\mathcal{A}$ is the finite set of actions. $P : \mathcal{S} \times \mathcal{A} \to \Delta_{\mathcal{S}}$ is the transition probability function and $\Delta_{\mathcal{S}}$ denotes the space of probability distributions over $\mathcal{S}$. The function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ maps rewards to state-action pairs. $D \in \Delta_{\mathcal{S}}$ is the initial state distribution and $\gamma \in [0, 1)$ the discount factor. The decision agent is modeled as a stationary policy $\pi : \mathcal{S} \to \Delta_{\mathcal{A}}$, where $\pi(a|s)$ is the probability of taking action a in state s. When a deterministic policy is required we simply take $a = \arg\max_{a'} \pi(a'|s)$. For simplicity, we will always write $a \sim \pi(\cdot|s)$ and according to which algorithm we are referring to it will be clear whether π is stochastic or deterministic. We parameterize π using a neural network with parameters $\boldsymbol{\theta} \in \Theta \subset \mathbb{R}^k$ and we write $\pi_{\boldsymbol{\theta}}$.

Given an MDP, we consider the human participants taking into account both egocentric and allocentric strategies when navigating (Alexander et al. 2020; Feigenbaum and Morris 2004). We define the state vector as $\boldsymbol{s} = \{x, y, \psi, \chi\}$, where x, y are coordinates with respect to a frame fixed to the environment and represent the allocentric capacities of the agent. Instead, ψ and χ are two categorical variables that describe the human egocentric behavior: the first tells the

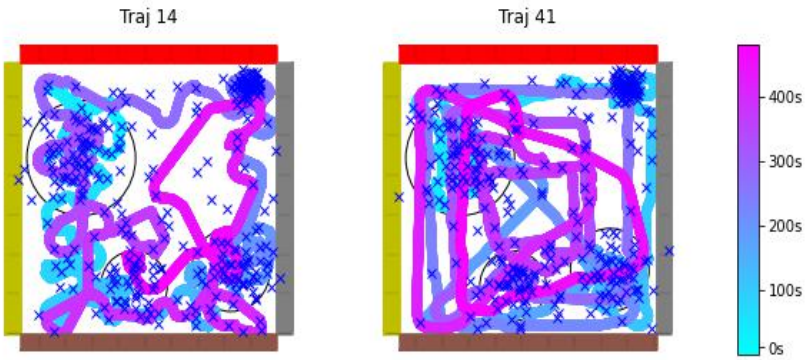

Figure 2. Two sample trajectories collected during the second day of tests. The bar on the right shows the time in seconds. The full set of the 50 trajectories is available in the supplementary materials.

agent whether it can see a coin or not in its vicinity, $\psi \in \{$ see coin, no coins $\}$, the second describes the "greedy" direction, i.e., the direction of the closest coin the agent has in its view, $\chi \in \{$east, northeast, north, northwest, west, southwest, south, southeast, no coins$\}$.

The artificial agent perceives the state and takes an action to interact with the environment. For computational reasons we discretize the x, y coordinates on a fine grid of $1m \times 1m$ and define the action space as $a \in \{$east, northeast, north, northwest, west, southwest, south, southeast$\}$. The transition to the next state always occurs deterministically in the direction of the action a taken by the agent. As convention in Fig. 2, north means going from bottom to top, south from top to bottom, east is left to right and west vice versa. The categorical state ψ stays 0 all the time unless there is a coin in a radius of $8m$ distance, when ψ turns 1 then also χ turns from "no coins" to one of the other directions. This is aligned with the original experiment where each coin pops-up when the human is at $8m$ from it. Finally, the rewards are simply represented by the coins in the environment where $r(\boldsymbol{s}, a) = 1$ for each coin collected. As in the original experiment, the agents automatically collect the reward once at $3m$ and D is a uniform distribution over $\mathcal{S}$.

Imitation Learning

Given a task and an agent performing the task, IL infers the underlying agent distribution via a set of an agent's demonstrations (state-action samples). Assuming the agent's behavior is parameterized by a NN with optimal parameters $\boldsymbol{\theta}^*$, we refer to the process of estimating $\boldsymbol{\theta}^*$ through a finite sequence of agent's demonstrations $\tau = (\boldsymbol{s}_{0:T}, a_{0:T})$ with $2 \leq T < \infty$ as IL. One way to formulate this problem is through maximum likelihood estimation:

$$\max_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}), \tag{1}$$

where $\mathcal{L}(\boldsymbol{\theta})$ denotes the log-likelihood and is equivalent to the logarithm of the joint probability of generating the expert demonstrations $\tau = \{\boldsymbol{s}_0, a_0, \boldsymbol{s}_1, a_1, \dots, \boldsymbol{s}_T, a_T\}$, i.e.,

$$\mathcal{L}(\boldsymbol{\theta}) = \log \mathbb{P}_D^{\boldsymbol{\theta}}(\tau). \tag{2}$$

$\mathbb{P}_D^{\boldsymbol{\theta}}(\tau)$ in (2) is defined as

$$\mathbb{P}_D^{\boldsymbol{\theta}}(\tau) = D(\boldsymbol{s}_0)\Big[\prod_{t=0}^{T} \pi_{\boldsymbol{\theta}}(a_t|\boldsymbol{s}_t)\Big]\Big[\prod_{t=0}^{T-1} P(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, a_t)\Big]. \tag{3}$$

Computing the logarithm of (3) and neglecting the elements not parameterized by $\boldsymbol{\theta}$ we obtain the following maximization problem

$$\max_{\boldsymbol{\theta}} \sum_{t=0}^{T} \log\big(\pi_{\boldsymbol{\theta}}(a_t|\boldsymbol{s}_t)\big). \tag{4}$$

Solving the maximization problem in Eq. (4) is the main objective of our IL step.

Reinforcement Learning

After defining a model, collecting the data and performing the imitation step, our final goal is to further refine the imitated policies using RL. In RL, the artificial agents are allowed to experience the task themselves and receive a reinforcement according to the reward function $r(\boldsymbol{s}_t, a_t)$. Mathematically, the goal is to find the policy parameters $\boldsymbol{\theta}$ which maximize the expected total discounted reward $J(\boldsymbol{\theta}) = \mathbb{E}_\tau[\sum_{t=0}^{\infty} \gamma^t r(\boldsymbol{s}_t, a_t)]$, where, as previously, $\tau = (\boldsymbol{s}_0, a_0, \boldsymbol{s}_1, a_1, \dots)$ is sampled according to $\boldsymbol{s}_0 \sim D$, $a_t \sim \pi_{\boldsymbol{\theta}}(\cdot|\boldsymbol{s}_t)$ and $\boldsymbol{s}_{t+1} \sim P(\cdot|\boldsymbol{s}_t, a_t)$. Our focus is on model-free RL methods in which the artificial agent does not know the transition probability function $P(\cdot|\boldsymbol{s}_t, a_t)$, and it can only explore the environment and experience rewards. Among these types of algorithms, we can distinguish two main groups: (i) algorithms that update the current policy

following the agent's generated trajectories according to this policy, also known as *on-policy* algorithms, and (ii) algorithms that update the current policy using experience from multiple policies used previously, known as *off-policy*. We provide a more thorough introduction on this difference in the supplementary materials.

State-of-the-art on-policy algorithms include Trust Region Policy Optimization (TRPO) (Schulman et al. 2015), and some robust variants such as Uncertainty Aware TRPO (UATRPO) (Queeney et al. 2021), and Proximal Policy Optimization (PPO) (Schulman et al. 2015). Whereas, off-policy methods include the Soft-Actor Critic (SAC) and Twin Delayed Deep Deterministic Policy Gradient (TD3) (Haarnoja et al. 2018; Fujimoto et al. 2018). All the mentioned algorithms are tested in our experiments and more details about the NNs design are available in the supplementary materials. It is worth noting that these algorithms are actor critic-based approaches, comprising both a policy network (the actor) and a critic network. In this context, we exclude value function-based approaches that rely solely on critic networks. This decision is motivated by the fact that the IL step returns only pre-trained policy networks, since the pre-training of critic networks is prevented by the absence of expert rewards (Levine et al. 2020).

Results

In this section, we present our results and describe all the steps that lead to our design. All the code and data to replicate the experiments are freely accessible at our GitHub repository*. An overview of the NNs used to parameterize $\pi_{\boldsymbol{\theta}}$ and all the hyperparameters used for each of the IL and RL algorithms are in the supplementary materials.

Pre-processing

Our first step is to collect and process the 50 trajectories, of 8 minutes each, recorded on the second day of tests. Each 8-minute trajectory consists of 28973 data points, on average, which means we collect a data point every $0.017s$, where the data points are the human agent's coordinates with respect to the fixed environment frame. Note that it is possible that a human agent does not move for a few seconds, for example, and then makes many rapid decisions about where to explore in the next milliseconds following this stationary period. Therefore, the first "few seconds" could be aggregated in a single data point while the next "milliseconds" would require more than a single point. As a result, we aggregate the data points considering the discretization of the (x, y) coordinates. After that, we go over each of the trajectories and determine the human decisions (i.e., for each aggregated state, the direction of the human's next movement). We cast each human decision for each trajectory in the pre-determined action space $\mathcal{A}$ and construct in this way our state-action pairs (i.e., actions a taken at state $\boldsymbol{s}$). This process allows us to reduce the average length of human trajectories from 28973 to 3464 data points without losing key information. Note that this processing is an expensive but necessary step for reducing the computational burden and enabling learning. Future research will focus on how to automate this step and developing methods which can handle learning from raw data.

Imitation Learning

We perform IL on each human trajectory individually rather than considering a single data set with all the trajectories. There are two main factors contributing to this choice. Firstly, as demonstrated in Figure 2, each human trajectory extensively covers the environment, rendering each trajectory inherently informative about the task at hand. Secondly, the policies generating these trajectories exhibit different distributions, leading to high variance in a single aggregated dataset. This variance ultimately undermines the effectiveness of the IL step as the distribution which better fits the aggregated data is close to the uniform distribution. The results of the IL step and all the details on the evaluation are illustrated, for 5 humans' trajectories, in Fig 3. In summary, we achieve good learning performance for several trajectories but not enough to match the human participants. A figure showing the IL performance for all the 50 trajectories is available in the supplementary materials.

Reinforcement Learning

We consider the 50 policies learnt from the 50 human trajectories during the IL step. We refine these policies using RL. We design the experiment as follows:

1. First, in order to determine which RL algorithm is more suitable for our goal, we take the same single policy learnt during the IL step and use it as initialization of each of the RL algorithms.
2. Given the cardinality of the state-space ($160 \times 160 \times 2 \times 9$), we consider 10^7 steps for performing RL. This means that the learning agent can leverage interactions that are on the order of 20 times the number of states.
3. As in the IL step, for each RL algorithm we run the learning process for 8 random seeds.

*https://github.com/VittorioGiammarino/Learning-from-humans-combining-imitation-and-deep-on-policy-reinforcement-learning-to-accomplish-su

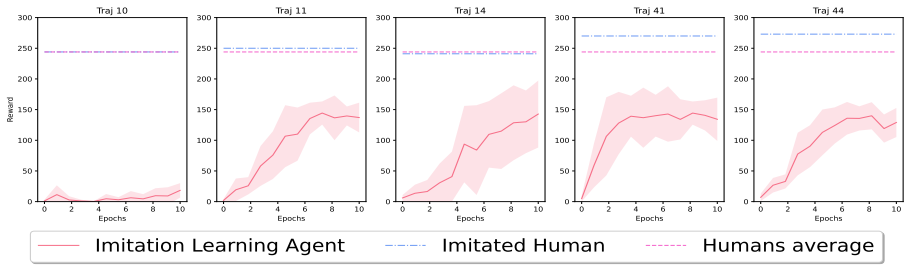

Figure 3. The results from the IL step for 5 different human trajectories are illustrated. For each human trajectory we solve the IL problem for 8 different seeds. For each seed, after every epoch we evaluate the performance of the learnt policy for 10 trials, each consisting of 3464 steps. The reported results for the "Imitation Learning Agent" show the reward averaged over 10 trials and 8 seeds; the shaded area shows the standard deviation over the seeds. The performance of the human trajectory used as data set for the imitation is labelled as "Imitated Human". "Humans average" is the average performance of the 50 human trajectories.

4. During the learning process, we evaluate the policy learnt every $30,000$ steps on 10 trials of 3464 steps each. We report averaged results over the 10 trials and 8 seeds. The shaded area in Fig. 4 and 5 shows the standard deviation over seeds.
5. After determining the most suitable RL algorithm, we rerun the whole experiment for 50 times in which each of the 50 policies learnt during the IL step is used as initialization of the selected RL algorithm.

Fig. 4 compares the various RL algorithms. PPO outperforms all other methods. Broadly speaking, on-policy algorithms, i.e., PPO, TRPO, and UATRPO, learn more effectively from a pre-initialized policy with respect to the off-policy algorithms TD3 and SAC. Refer to the supplementary materials for more details.

Consequently, we proceed by combining IL together with PPO and compare it with the PPO-only alternative. Fig. 5 illustrates the final results for the same trajectories of Fig. 3 and a figure showing this final result for all the 50 trajectories is available in the supplementary materials. In summary, IL followed by PPO (IL+PPO) outperforms the average human performance and its imitated expert, 39 (78%) and 31 (62%) times, respectively, over the 50 human trajectories. On the other hand, the PPO-only alternative cannot get close to these results in 10^7 steps. Table 1 summarizes the comparison between humans and IL+PPO policies with respect to the total amount of collectable rewards.

Note that, Fig. 5 provides interesting insights on why pre-training with IL makes sense in foraging tasks with sparse rewards. We observe that, in addition to a different initial performance, the IL+PPO and the PPO-only agents show really different exploration strategies which lead to a different reward convergence rate (the difference in rates is clearly visible in Fig. 5). The exploration strategy used in PPO-only follows the original approach presented in Schulman et al. (2017), where an entropy regularization term is incorporated in the policy gradient step in order to encourage stochasticity in the decision policy. In this setup, the human-inspired exploration strategy of IL+PPO represents the main strength of the method and the main source of difference with the PPO-only agents.

Robustness to reward distribution shift and the importance of egocentric representations

In this section, we test the learnt policies for robustness to a reward distribution shift. The motivation is to explore how quickly the artificial agents grasp changes in the environment and adapt to these changes. As in the original experiment, 325 coins are placed across the environment; however, this time according to the new distribution in Fig. 6a, we include the original coins distribution in Fig. 6b to facilitate the comparison with Fig. 6a. Specifically, 50 coins are distributed according to $\mathcal{N}((-70, 30), 5^2\mathbf{I})$, 75 according to $\mathcal{N}((60, -20), 11^2\mathbf{I})$, 100 according to $\mathcal{N}((-40, 45), 15^2\mathbf{I})$ and 100 according to $\mathcal{N}((0, 60), 13^2\mathbf{I})$.

We design the experiment similarly to the RL study and the IL+PPO experiments in Fig. 4 and Fig. 5. Overall, we run, for 8 different random seeds, 100 learning experiments of 2×10^6 steps each, where in the first 50 we initialize using the policies learnt with only IL (Fig. 3), while, in the second 50, we initialize using the policies learnt by IL+PPO (Fig. 5). The results are summarized in Table 2 and show that the policies learnt using both IL+PPO generalize well to novel reward distributions. The figures showing the detailed 100 experiments are available in the supplementary materials.

In order to produce these results, we conclude that, given the state vector representation as $\boldsymbol{s} = \{x, y, \psi, \chi\}$, the RL agents and their exploration strategies must heavily rely on egocentric information, i.e., the variables ψ and χ. This would explain the algorithm performance in the novel reward environment in Fig. 6a, where the previously learnt allocentric representation is no longer informative. On the other hand, a

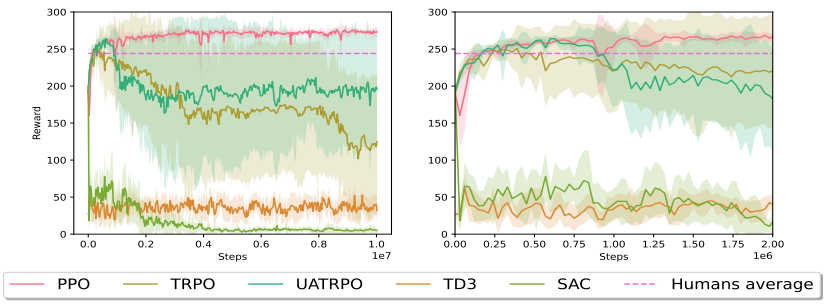

Figure 4. The results for the RL algorithms initialized with the same policy are illustrated. For each algorithm we run the learning process for 10^7 steps and 8 different random seeds. For each seed, after every $30,000$ steps we evaluate the performance of the learnt policy for 10 trials each of 3464 steps. The reported results show the reward averaged over 10 trials and 8 seeds, the shaded area shows the standard deviation over seeds.

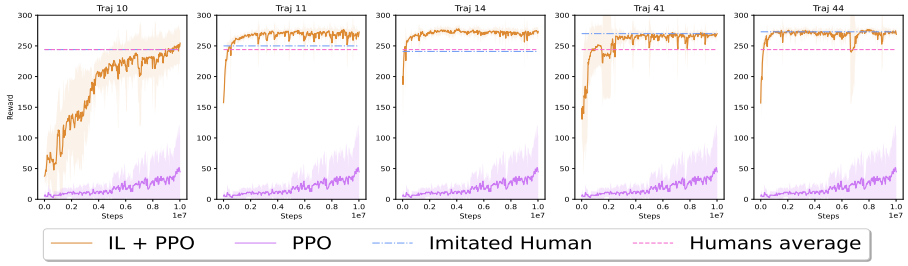

Figure 5. The results of the IL+PPO method compared with the PPO-only alternative, the average human performance and the performance of the imitated expert are illustrated for the trajectories of Fig. 3. The experiment design and the reported results follow the same criterion as in Fig. 4.

Table 1. A summary of the comparison between IL+PPO and human performance in the original experiment. The table shows the percentage out of 50 learnt policies, for both the IL+PPO and the human agents, where the agent collected at least a certain percentage of rewards. As an example, the table is showing that, for the IL+PPO method, 19 policies (38%) can collect at least 260 coins ($> 80\%$). We compare these results with the human performance where each trajectory is considered as a single human policy. Note that, the IL+PPO policies perform better than humans on average but cannot do better than the best participants.

	Performance Lower Bound			
	$> 70\%$	$> 80\%$	$> 85\%$	$> 90\%$
IL+PPO	88%	38%	0%	0%
Humans	80%	22%	10%	0%

(a) New rewards distribution.

(b) Old rewards distribution.

Figure 6. Fig. 6a shows the new reward distribution that was not previously seen by any of the artificial agents. We include the previous reward distribution in Fig. 6b to facilitate the comparison.

Table 2. A summary of the results for the rewards distribution shift experiment. The table shows the fraction out of 50 policies, for each initialization method, where after learning for 2×10^6 steps, the agent is able to collect at least a certain percentage of rewards. As an example, the table is showing that, for the IL+PPO initialization, 46 policies (92%) can collect at least 270 coins ($> 80\%$) in this new scenario after learning for only 2×10^6 steps.

	Performance Lower Bound			
	$> 70\%$	$> 80\%$	$> 90\%$	$> 95\%$
IL-only initialization	28%	18%	0%	0%
IL+PPO initialization	98%	92%	18%	0%

strategy based on egocentric exploration which facilitates the generation of a new allocentric representation of the environment would explain the results in Table 2. In other words, we suggest that our IL+PPO algorithms exhibit coding of behavioral variables analogous to the observation in animals (Alexander et al. 2020), where electrophysiological recording during foraging strategies indicate neural coding in both egocentric and allocentric coordinate frames.

To demonstrate the veracity of this claim we rerun the entire set of experiments, which includes IL as in Fig. 3 and IL+PPO as in Fig. 5, but this time only providing allocentric information to the artificial agents. In other words, we reduce the state vector from $\boldsymbol{s} = \{x, y, \psi, \chi\}$ to $\boldsymbol{s} = \{x, y\}$. The results for a selected number of human trajectories are summarized in Fig 7. For the entire set of trajectories refer to the supplementary materials. The final results show that agents trained with the full state $\boldsymbol{s} = \{x, y, \psi, \chi\}$ outperform agents trained with the allocentric only state $\boldsymbol{s} = \{x, y\}$ 74% of the times for IL (37 out of 50) and 100% of the times for IL+PPO. We conclude that our learnt policies heavily rely on egocentric data and that the absence of such information compromises to a large extent the learning performance as illustrated in Fig. 7.

Discussion

In this paper, 50 human navigation trajectories were collected in a virtual open-field environment. We extracted a navigation control policy from each of these trajectories and introduced an MDP setting to capture the navigational human decision making. We learned policies consistent with the experimental data using imitation learning based on log-likelihood maximization for each of the trajectories.

After obtaining a control policy for each trajectory, we used all of them as a starting point for RL, seeking to find policies that can efficiently outperform the human participants in the same experimental setting. We tested state-of-art on-policy (PPO, TRPO, UATRPO) and off-policy (TD3, SAC) algorithms. We explained more extensively how these two categories differ in the supplementary materials. Briefly, the main element of difference lies in the data used to update the policy network $\pi_{\boldsymbol{\theta}}$ and in how we compute and approximate the critic network. Off-policy algorithms are usually faster to converge but introduce a large bias in the critic estimate, which results in more oscillatory learning which often jeopardizes the IL initialization. On the other hand, the tested on-policy algorithms are more conservative, and the optimization step is constrained so not to diverge too much from the current policy $\pi_{\boldsymbol{\theta}}$. This results in a slower but more steady improvement of performance. Our preference towards PPO with respect to the other on-policy algorithms is the result of empirical experiments which corroborates other well-known empirical studies on the matter (Andrychowicz et al. 2020; Engstrom et al. 2020).

Finally, we examined the sensitivity of the IL+PPO and IL-only policies to a different reward distribution and investigated to what extent our artificial agents rely on egocentric information. The final results showed that learning only from data is not enough to match human performance and does not lead to robustness over the reward distribution (Fig. 3 and Table 1). On the other hand, IL followed by PPO (IL+PPO) showed impressive results in the original experiment and it led to good generalization of the task (Fig. 5 and Table 2). Further, we showed that such results are associated with the use of egocentric information, which are crucial in enhancing learning performance both in the IL and the IL+RL setting when compared with the use of allocentric information alone.

In summary, we have developed a method to learn bio-inspired policies from human navigation data, which can be further refined to achieve human-level performance. This approach to modeling human navigational policies can be of great utility for aerial and ground unmanned navigation tasks including scientific exploration and search and rescue operations.

Acknowledgements

Research was partially supported by the NSF under grants CFF-2200052, IIS-1914792, DMS-1664644 and NSF-1829398, by the ONR under grants N00014-19-1-2571 and N00014-21-1-2844, by the ONR DURIP under grants N00014-17-1-2304, by the NIH under grants R01 GM135930 and UL54 TR004130, and by the Boston University Kilachand Fund for Integrated Life Science and Engineering.

References

Abbeel P, Coates A and Ng AY (2010) Autonomous helicopter aerobatics through apprenticeship learning. *The International Journal of Robotics Research* 29(13): 1608–1639.

Abbeel P and Ng AY (2004) Apprenticeship learning via inverse reinforcement learning. In: *Proceedings of the twenty-first International Conference on Machine Learning*. p. 1.

Alexander AS, Carstensen LC, Hinman JR, Raudies F, Chapman GW and Hasselmo ME (2020) Egocentric boundary vector tuning of the retrosplenial cortex. *Science Advances* 6(8): eaaz2322.

Andrychowicz M, Raichuk A, Stańczyk P, Orsini M, Girgin S, Marinier R, Hussenot L, Geist M, Pietquin O, Michalski M et al. (2020) What matters in on-policy reinforcement learning?

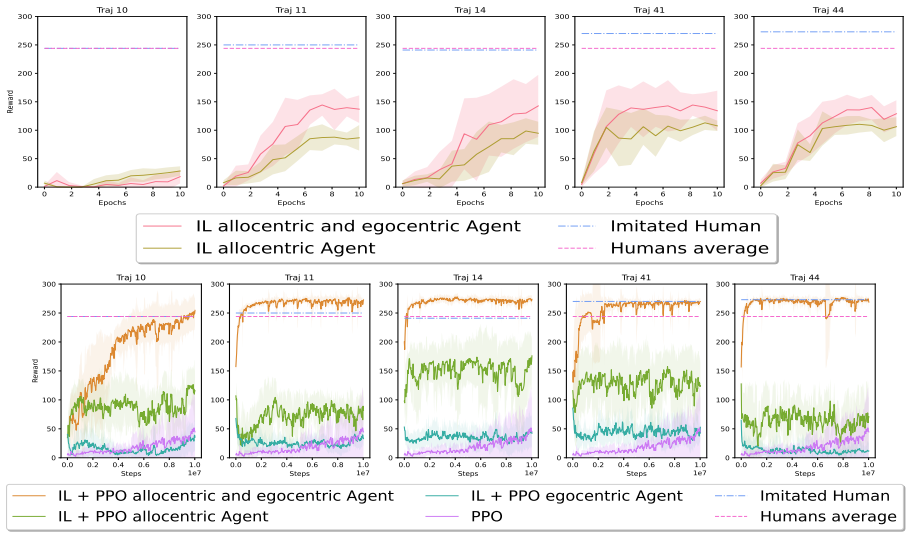

Figure 7. Performance comparison, for selected trajectories, among agents trained with full state including both egocentric and allocentric information $\boldsymbol{s} = \{x, y, \psi, \chi\}$ versus an allocentric only state $\boldsymbol{s} = \{x, y\}$. The comparison is done for both IL (upper figure) and IL+PPO (lower figure). For completeness, we also show the PPO agent without IL initialization performance (PPO allocentric and egocentric Agent).

a large-scale empirical study. *arXiv preprint arXiv:2006.05990* .

Botvinick M, Ritter S, Wang JX, Kurth-Nelson Z, Blundell C and Hassabis D (2019) Reinforcement learning, fast and slow. *Trends in Cognitive Sciences* 23(5): 408–422.

Cheng C, Yan X, Wagener N and Boots B (2019) Fast policy learning through imitation and reinforcement. In: *Uncertainty in Artificial Intelligence*.

Dulac-Arnold G, Mankowitz D and Hester T (2019) Challenges of real-world reinforcement learning. *arXiv preprint arXiv:1904.12901* .

Engstrom L, Ilyas A, Santurkar S, Tsipras D, Janoos F, Rudolph L and Madry A (2020) Implementation matters in deep policy gradients: A case study on ppo and trpo. In: *International Conference on Learning Representations*.

Feigenbaum JD and Morris RG (2004) Allocentric versus egocentric spatial memory after unilateral temporal lobectomy in humans. *Neuropsychology* 18(3): 462.

Finn C, Levine S and Abbeel P (2016) Guided cost learning: Deep inverse optimal control via policy optimization. In: *International Conference on Machine Learning*. PMLR, pp. 49–58.

Fujimoto S, Hoof H and Meger D (2018) Addressing function approximation error in actor-critic methods. In: *International Conference on Machine Learning*. PMLR, pp. 1587–1596.

Ghasemipour SKS, Zemel R and Gu S (2020) A divergence minimization perspective on imitation learning methods. In: *Conference on Robot Learning*. PMLR, pp. 1259–1277.

Goddu MK, Lombrozo T and Gopnik A (2020) Transformations and transfer: Preschool children understand abstract relations and reason analogically in a causal task. *Child Development* 91(6): 1898–1915.

Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A and Bengio Y (2020) Generative adversarial networks. *Communications of the ACM* 63(11): 139–144.

Gopnik A, Griffiths TL and Lucas CG (2015) When younger learners can be better (or at least more open-minded) than older ones. *Current Directions in Psychological Science* 24(2): 87–92.

Haarnoja T, Zhou A, Abbeel P and Levine S (2018) Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: *International Conference on Machine Learning*. PMLR, pp. 1861–1870.

Hester T, Vecerik M, Pietquin O, Lanctot M, Schaul T, Piot B, Horgan D, Quan J, Sendonaris A, Osband I et al. (2018) Deep q-learning from demonstrations. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Ho J and Ermon S (2016) Generative adversarial imitation learning. *Advances in Neural Information Processing Systems* 29.

Jones SS (2009) The development of imitation in infancy. *Philosophical Transactions of the Royal Society B: Biological Sciences* 364(1528): 2325–2335.

Kang B, Jie Z and Feng J (2018) Policy optimization with demonstrations. In: *International Conference on Machine Learning*. PMLR, pp. 2469–2478.

Kober J, Mohler B and Peters J (2010) Imitation and reinforcement learning for motor primitives with perceptual coupling. *From motor learning to interaction learning in robots* : 209–225.

Levine S, Kumar A, Tucker G and Fu J (2020) Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643* .

Libardi G, De Fabritiis G and Dittert S (2021) Guided exploration with proximal policy optimization using a single demonstration. In: *International Conference on Machine Learning*. PMLR, pp. 6611–6620.

Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D and Riedmiller M (2013) Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* .

Moore K, Yi C, Dunne M, Stern C and McGuire J (2021) Virtual human foraging behavior follows predictions for heavy-tailed search. *In Society for Neuroscience* Online.

Nair A, McGrew B, Andrychowicz M, Zaremba W and Abbeel P (2018) Overcoming exploration in reinforcement learning with demonstrations. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 6292–6299.

Offerman T and Sonnemans J (1998) Learning by experience and learning by imitating successful others. *Journal of Economic Behavior & Organization* 34(4): 559–575.

Otte M, Correll N and Frazzoli E (2013) Navigation with foraging. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 3150–3157.

Pomerleau DA (1991) Efficient training of artificial neural networks for autonomous navigation. *Neural Computation* 3(1): 88–97.

Queeney J, Paschalidis IC and Cassandras CG (2021) Uncertainty-aware policy optimization: A robust, adaptive trust region approach. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35. pp. 9377–9385.

Rajeswaran A, Kumar V, Gupta A, Vezzani G, Schulman J, Todorov E and Levine S (2017) Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087* .

Ratliff ND, Bagnell JA and Zinkevich MA (2006) Maximum margin planning. In: *Proceedings of the twenty-third International Conference on Machine learning*. pp. 729–736.

Ross S and Bagnell D (2010) Efficient reductions for imitation learning. In: *Proceedings of the thirteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, pp. 661–668.

Ross S and Bagnell JA (2014) Reinforcement and imitation learning via interactive no-regret learning. *arXiv preprint arXiv:1406.5979* .

Ross S, Gordon G and Bagnell D (2011) A reduction of imitation learning and structured prediction to no-regret online learning. In: *Proceedings of the fourteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, pp. 627–635.

Ruggeri A, Pelz M, Gopnik A and Schulz E (2021) Toddlers search longer when there is more information to be gained. *PsyArXiv preprint* .

Schaal S (1999) Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences* 3(6): 233–242.

Schulman J, Levine S, Abbeel P, Jordan M and Moritz P (2015) Trust region policy optimization. In: *International Conference on Machine Learning*. PMLR, pp. 1889–1897.

Schulman J, Wolski F, Dhariwal P, Radford A and Klimov O (2017) Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* .

Scone S and Phillips I (2010) Trade-off between exploration and reporting victim locations in usar. In: *2010 IEEE International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*. IEEE, pp. 1–6.

Serrano-Cuevas J, Morales EF and Hernández-Leal P (2020) Safe reinforcement learning using risk mapping by similarity. *Adaptive Behavior* 28(4): 213–224.

Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, Hubert T, Baker L, Lai M, Bolton A et al. (2017) Mastering the game of go without human knowledge. *Nature* 550(7676): 354–359.

Subramanian K, Isbell Jr CL and Thomaz AL (2016) Exploration from demonstration for interactive reinforcement learning. In: *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*. pp. 447–456.

Sun W, Bagnell JA and Boots B (2018) Truncated horizon policy search: combining reinforcement learning and imitation learning. In: *International Conference on Learning Representations*.

Sutton RS and Barto AG (2018) *Reinforcement learning: An introduction*. MIT press.

Syed U and Schapire RE (2010) A reduction from apprenticeship learning to classification. *Advances in Neural Information Processing Systems* 23.

Uchendu I, Xiao T, Lu Y, Yan M, Simón JLP, Bennice M, Fu C and Hausman K (2021) Demonstration-guided q-learning. *NIPS Workshop on Robot Learning: Self-Supervised and Lifelong Learning* .

Uchendu I, Xiao T, Lu Y, Zhu B, Yan M, Simon J, Bennice M, Fu C, Ma C, Jiao J et al. (2022) Jump-start reinforcement learning. *arXiv preprint arXiv:2204.02372* .

Vecerik M, Hester T, Scholz J, Wang F, Pietquin O, Piot B, Heess N, Rothörl T, Lampe T and Riedmiller M (2017) Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *arXiv preprint arXiv:1707.08817* .

Walker CM, Williams JJ, Lombrozo T and Gopnik A (2012) Explaining influences children's reliance on evidence and prior knowledge in causal induction. In: *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 34.

Ziebart BD, Maas AL, Bagnell JA, Dey AK et al. (2008) Maximum entropy inverse reinforcement learning. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 8. Chicago, IL, USA, pp. 1433–1438.