What Do You Want Me To Do? Addressing Model Differences for Human-Aware Decision-Making from A Learning Perspective

by

Ze Gong

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved July 2022 by the
Graduate Supervisory Committee:

Yu Zhang, Chair
Subbarao Kambhampati
Heni Ben Amor
Wenlong Zhang

ARIZONA STATE UNIVERSITY

August 2022

ABSTRACT

As intelligent agents become pervasive in our lives, they are expected to not only achieve tasks alone but also engage in tasks with humans in the loop. In such cases, the human naturally forms an understanding of the agent, which affects his perception of the agent's behavior. However, such an understanding inevitably deviates from the ground truth due to reasons such as the human's lack of understanding of the domain or misunderstanding of the agent's capabilities. Such differences would result in an unmatched expectation of the agent's behavior with the agent's optimal behavior, thereby biasing the human's assessment of the agent's performance. In this dissertation, I focus on when these differences are due to a biased belief about domain dynamics. I especially investigate the impact of such a biased belief on the agent's decision-making process in two different problem settings from a learning perspective. In the first setting, the agent is tasked to accomplish a task alone but must infer the human's objectives from the human's feedback on the agent's behavior in the environment. In such a case, the human biased feedback could mislead the agent to learn a reward function that results in a sub-optimal and, potentially, undesired policy. In the second setting, the agent must accomplish a task with a human observer. Given that the agent's optimal behavior may not match the human's expectation due to the biased belief, the agent's optimal behavior may be viewed as inexplicable, leading to degraded performance and loss of trust. Consequently, this dissertation proposes approaches that (1) endow the agent with the ability to be aware of the human's biased belief while inferring the human's objectives, thereby (2) neutralize the impact of the model differences in a reinforcement learning framework, and (3) behave explicably by reconciling the human's expectation and optimality during decision-making.

ACKNOWLEDGMENTS

When I was a master student, I have never envisioned to pursue a Ph.D. degree. At SBU, that was the first time I know about AI. I felt like it opened a gate to a whole new world for me. Later, I got the opportunity to work with Prof. Luis Ortiz on my first ever AI project from which I learned a lot. After then, I started to have strong interests on AI research and decided to pursue a Ph.D. degree.

Pursuing a Ph.D. degree is always not easy and challenging. First of all, I would like to thank my advisor Prof. Yu Zhang. He has been providing me thoughtful, sincere advises and continuous supports on research, my future career and my life, especially when I was struggling in the darkest time during the pandemic. In addition, I would like to thank my committee members, Prof. Subbarao Kambhampati, Prof. Heni Ben Amor, Prof. Wenlong Zhang. They provided me valuable and insightful comments and suggestion that inspired me to improve my work and learned to be a better researcher. I would also want to thank my lab mates, Mehrdad Zakershahrak, Akkamahadevi Hanni, Andrew Boateng, Akshay Sharma, with their kindest help in my work and life. Moreover, I am very grateful to have my friends, Zhe Wang, Xi Yang, Xin Ye, Rui Zhang for all the cheerful and relaxing moments we have. They brightened my life in those stressful days.

Most importantly, I want to thank my dearest parents from the bottom of my heart. Without them, I would not become who I am. I know it is not easy for them to support me study abroad. They always show the greatest love to me. They are the most supportive people behind me and always encourage me to pursue my goals. Unfortunately, my mom passed away one year ago, and I feel very regretful that I couldn't get back to stay with her in her last days. Hope I can complete her wishes and always make her proud! Rest in peace, my dearest mom! I love you forever!

TABLE OF CONTENTS

Page

LIST OF TABLES vi

LIST OF FIGURES vii

CHAPTER

1 INTRODUCTION 1

1.1 Problem Statement 1

1.2 Related Work 7

1.2.1 Learning from Human Feedback 7

1.2.2 Model Asymmetry and Reconciliation in A Human-Agent Team 8

1.3 Dissertation Outline 10

2 GENERALIZED REWARD LEARNING WITH BIASED BELIEFS ABOUT DOMAIN DYNAMICS 11

2.1 Introduction 11

2.2 Our Approach 13

2.2.1 Problem Formulation 14

2.2.2 Methodology 16

2.3 Evaluation 21

2.3.1 Simulated Navigation Domain 22

2.3.2 User Study with the Coffee Robot Domain 26

2.4 Conclusion 31

3 NEUTRALIZING THE IMPACT OF BIASED BELIEF IN PREFERENCE-BASED REINFORCEMENT LEARNING 34

3.1 Introduction 34

CHAPTER Page

3.2 Related Work 38

3.3 Our Approach 39

3.3.1 Preliminaries 39

3.3.1.0.1 Deep Reinforcement Learning from Human Preferences 40

3.3.1.0.2 PEBBLE 41

3.3.2 Problem Formulation 41

3.3.3 Methodology 42

3.3.3.1 Inferring Reward function and Human Belief Simultaneously 43

3.3.3.2 Human Preference Model 46

3.3.3.3 Queries Selection 46

3.3.3.4 Algorithm Overview 48

3.4 Evaluation 49

3.4.1 Experiments Setups 50

3.4.1.1 Learn from Simulated Human Models 50

3.4.1.2 Baselines 50

3.4.2 Benchmark Tasks 51

3.4.2.1 2D Navigation Task 51

3.4.2.2 Lunar Lander Task 52

3.4.2.3 Locomotion Tasks 54

3.4.3 Results and Discussions 58

3.4.3.1 Effects of Different Priors 59

3.4.3.2 Explore Different Query Sampling Methods 62

CHAPTER Page

3.5 Conclusion 62

4 EXPLICABLE POLICY SEARCH 64

4.1 Introduction 64

4.2 Related Work 69

4.3 Our Approach 71

4.3.1 Problem Formulation 71

4.3.2 Explicable Policy Search (EPS) 73

4.3.3 Surrogate Reward Function 74

4.4 Evaluation 79

4.4.1 Synthetic Navigation Domains 79

4.4.1.1 Baselines 81

4.4.1.2 Results and Discussion 82

4.4.1.3 Synthetic Experiments Without Biases 86

4.4.1.4 Sensitivity Analysis of the Reconciliation Hyperparameter 87

4.4.2 Autonomous Driving Domain 87

4.4.2.1 Results and Discussion 89

4.5 Conclusion 94

5 CONCLUSION 96

REFERENCES 100

LIST OF TABLES

Table Page

1. Comparison of GeReL, SERD, GeReL^{-}, and MaxEnt-IRL for the Two Settings in Our Simulation with Respect to the L_2 Distance between the Estimated Values and the Ground Truth. The Third Column (I.e., $d(\pi)$) Shows the KL Divergence between the Estimated Human's Expectation of the Agent's Softmax Policy and that under the Ground Truth. 26
2. Averaged Participants' Responses to the Two Questions for Each Setting before Viewing the Demonstrations. They Are Asked in a 5-Point Likert Scale Where 1 Is the Lowest and 5 the Highest. 30
3. Comparison of EPS to Baselines Using Averaged Return. 84
4. Comparison of EPS to Baselines in Terms of Explicability Score. 84
5. Comparison between EPS and Baselines on an Autonomous Driving Domain. 91

LIST OF FIGURES

Figure Page

1. Human Biased Belief about the Agent's Domain Dynamics in a Human-Agent Team. 2
2. The Surrogate Reward Function Learned from the Human Feedback Contains Information of Objectives Together with the Human Belief about Domain Dynamics. It Doesn't Accurately Describe the Objectives Solely and It Is a ‘‘Biased" Version of the Objectives Subject to the Human Belief about Domain Dynamics. 4
3. Workflow of GeReL. Using the Robot's True Transition Model, the Robot Randomly Generates a Set of Demonstrations Which Are Evaluated by the Human. The Human Is Assumed to Provide His Rating for Each Instance according to the Reward Function and His Belief about the Robot's Domain Dynamics. The Ratings Will Be Used to Update the Estimated Reward Function and Human's Understanding of the Robot. Gray Circles Denote the Latent Variables While the Observed Are in White. 14
4. Rewards Learned by Different Approaches. 23
5. Comparison of the Performance in Terms of Reward Learning among GeReL, SERD, GeReL^{-}, MaxEnt-IRL with the prior Also Shown. 24
6. Comparison of the Performance in Terms of Belief Learning among GeReL, SERD, GeReL^{-}, MaxEnt-IRL with the prior Also Shown. 25
7. Comparison of the Performance in Terms of Expectation Recovery among GeReL, SERD, GeReL^{-}, MaxEnt-IRL with the prior Also Shown. 25

Figure Page

8. The Coffee Robot Domain. The Weather Could Be Rainy or Sunny, and the Robot May Choose to Use an Umbrella or Operate in the Rain. We Use Two Types of Robots (Humanoid vs. Mobile) to Perform the Same Set of Demonstrations that Cover the Various Situations that May Occur. 27
9. The Introduction of the Tasks on MTurk. We Present the Appearance of One of the Two Types of Robots (Humanoid) to the Participants. 28
10. Questions for Eliciting the Paticipants' Belief and Rewards of the Domain and Task..... 28
11. One Demonstration of the Coffee Robot Is Illustrated to the Participants. They Need to Rate It Using the Slider Below. 29
12. Feature Weights Learned by GeReL and GeReL^{-}. 31
13. The Human User Observes the Robot's Behavior and Provides Preferences Based on Her Own Reward Model and Belief about the Agent's Domain Dynamics, Which May Be Different from the True Dynamics. Our Goal Is to Infer the Reward Function While Neutralize the Impact of such Bias, Thereby Obtain an Optimal Policy..... 35
14. Workflow of Our Approach. Intelligent Agent Collects Data by Interacting with Environment. We Then Select a Set of Trajectory Segments to Solicit the Human Preferences about the Agent's Behaviors Which Will Be Used to Learn a Surrogate Reward Function and Human Belief about Domain Dynamics. The Agent's Policy Is Optimized Using the Learned Reward Function..... 43
15. Illustration of the Domains We Test On. 53

Figure Page

16. Learning Curves on the Task Navigation 1 as Measured on the Ground Truth Reward. 55
17. Learning Curves on the Task Navigation 2 as Measured on the Ground Truth Reward. 56
18. Learning Curves on the Lunar Lander Task as Measured on the Ground Truth Reward. 56
19. Learning Curves on the Hopper Task as Measured on the Ground Truth Reward. 57
20. Learning Curves on the Cheetah Task as Measured on the Ground Truth Reward. 57
21. Trajectories Learned by PEBBLE and PrefBias Agent in 2D Navigation Domains. For Each Domain, the left Is for PrefBias Agent While the right Is for PEBBLE Agent. 58
22. L2 Error between Estimated Human Belief Model and Ground Truth on the Navigation Task. 60
23. L2 Error between Surrogate Reward Function and Ground Truth on the Navigation Task. 60
24. Learning Curves on the Navigation Task as Measured on the Ground Truth Reward. 61
25. Motivating Scenario to Demonstrate Explicable Trajectories. 65
26. Problem Setting of EPS. Shaded Nodes Are Known to the Agent and Unshaded Nodes Are Unknown. 66

Figure Page

27. Comparison of Different Learning Methods with Human's Expectation (left). The Dark Grey Area Represents the Walls. The Brown Area Is the Pit (with -100 Penalty) and the Green Area Is the Goal (with +100 Reward). The Blue Area Represents Icy Roads and the Yellow Area Represents Sandy Roads. 83
28. Visualization of the Learned Surrogate Reward Functions for D1-D4. The Darker the Lower Reward Value and the Brighter the Higher. 84
29. EPS Agent's Behaviors for Different Domains When There Are No Human Biases. 85
30. Comparison of the Learning Process of EPS and SAC in Terms of Return. . 86
31. EPS Agent's Behaviors with Different λ in All Domains. Light Green Trajectories Have Smaller λ Values While Dark Green Trajectories Have Larger λ Values. 88
32. Solicit Human's Preference of Several Important Feature regarding Driving Behaviors. 90
33. Scenario Introduction and Eliciting the Human Expectation-Based Preference. 91
34. EPS Agents' Behaviors in the Autonomous Driving Domain Illustrated in Three Characteristic Steps from the Top to Bottom for Each Agent. Its Average Rating and Standard Deviation Are ($\mu = 7.6, \sigma = 2.4$). 93
35. SAC Agents' Behaviors in the Autonomous Driving Domain Illustrated in Three Characteristic Steps from the Top to Bottom for Each Agent. Its Average Rating and Standard Deviation Are ($\mu = 5.0, \sigma = 2.3$)..... 93

Figure Page

36. DRLHP Agents' Behaviors in the Autonomous Driving Domain Illustrated in Three Characteristic Steps from the Top to Bottom for Each Agent. Its Average Rating and Standard Deviation Are ($\mu = 5.9, \sigma = 2.2$). 94

Chapter 1

INTRODUCTION

1.1 Problem Statement

Intelligent agents are quickly becoming parts of our daily lives in a variety of domains, including smart home, autonomous driving, entertainment, education and so on. In such domains, the agents are expected to perform in human inhabited environments and even collaborate closely with them, rather than accomplishing tasks alone. Similar as in a human-human team, the human always has a belief about her agent teammate's domain dynamics, together with her objectives by which an expectation of the agent's behavior is generated and it would influence the human's evaluation of the agent's task performance in a human-agent team. However, the human belief is inevitably biased (i.e., deviates from the ground truth domain dynamics as model differences), as shown in Figure 1, due to several factors, such as asymmetry in knowledge of the world, misunderstanding of the agent's capability, and human's bounded rationality. Such biased belief would potentially bring about an expectation that largely deviates from the agent's optimal policy and result in a wrong evaluation of the agent's behavior. For a human-in-the-loop learning task, the human may provide the agent with ratings, preferences, demonstrations, critiques and so on, from which the agent learns human rewards and optimizes policy. Moreover, the biases could come from human rewards, belief about the dynamics or computations in decision-making. In this dissertation, we would assume the human is noisily rational and investigate the impacts of model differences within the domain dynamics. We mainly focus

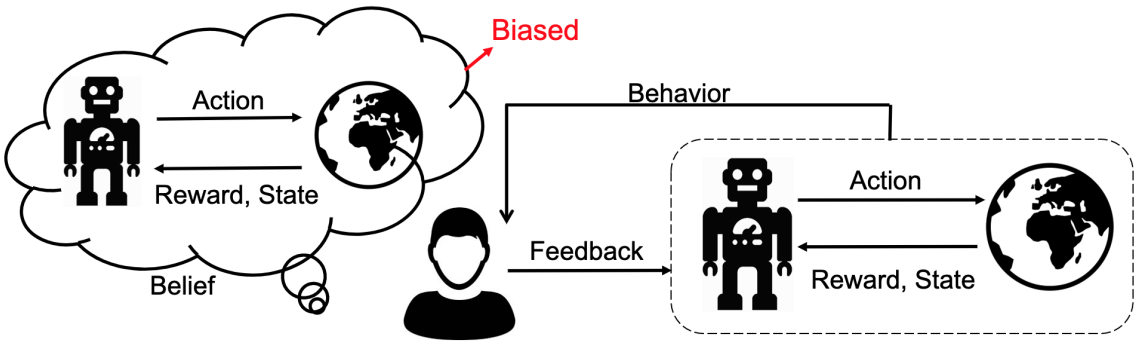


Figure 1. Human biased belief about the agent's domain dynamics in a human-agent team.

on three learning tasks, reward learning with biased belief about domain dynamics, preference-based reinforcement learning under biased belief, and policy optimization under human expectation subject to the biased belief about domain dynamics.

First of all, for a human-in-the-loop learning task, the agent is tasked to learn a reward function by soliciting human's feedback on its own behavior. In a human-agent team, the ability of the agent to understand the human's intent and preferences becomes a determinant for achieving effective human-agent teaming. Reinforcement learning agents usually learn by interacting with the environment and optimizing its behavior with respect to the received reward signals which is predefined by the system designer. However, manually designing a reward function that correctly conveys what the human really wants is challenging, especially in complex domains and tasks. Instead of relying on expensive and vulnerable reward engineering, the methods of learning a reward function using human data has been studied extensively before. Some researchers (Ng and Russell 2000) formulated this problem as an Inverse Reinforcement Learning (IRL) problem (Russell 1998). The reward function is recovered from optimal policies or behaviors demonstrated by human experts. Such expert demonstrations,

however, are often difficult to obtain in real-world tasks. To address this problem, learning methods based on non-expert user feedback of the agent's behavior, such as ratings and preferences, (Daniel et al. 2014; Dorsa Sadigh, Sastry, and Seshia 2017; Cui and Niekum 2018; Lee, Smith, and Abbeel 2021; Christiano et al. 2017; Wirth and Fürnkranz 2013c; Busa-Fekete et al. 2014) are developed and attracting more attentions. The intelligent agent demonstrates selected rollouts to the human to solicit human's feedback on them, from which the agent learns to optimize its policy that is consistent with the human's feedback.

A prevalent approach introduced in existing work is that we model the human feedback generation process with a surrogate reward function. This surrogate reward function is learned from the observed human feedback in a supervised learning framework. The agent then optimizes its policy by maximizing the discounted sum of this learned surrogate reward function. But, does this surrogate reward function accurately describe the human's objectives? Consider a teacher teaches something to a student. Even with an identical objectives, different belief about the student would make the teacher have different expectation on the student and teach in different ways. Similarly, in a human-agent team, the agent learn the task objectives from the human's feedback on its behaviors. The human usually provide feedback that contains information of a policy the human want the agent to follow (i.e., expectation of the agent's behavior). As we interpret that such expectation is determined by the human's true objectives and her belief about the domain dynamics, the learned surrogate reward function actually consist of information of both of them, and it cannot properly convey the objectives solely, especially when the belief about the domain dynamics is different from the ground truth, as shown in Figure 2.

A common assumption made implicitly in these prior works is that the human

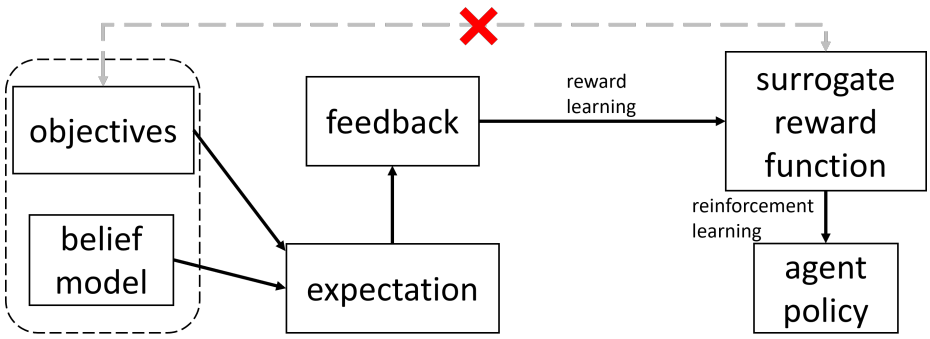


Figure 2. The surrogate reward function learned from the human feedback contains information of objectives together with the human belief about domain dynamics. It doesn't accurately describe the objectives solely and it is a "biased" version of the objectives subject to the human belief about domain dynamics.

always maintains a correct belief of the agent's domain dynamics, and the belief impact is overlooked in the learning process. This, however, may not be the case in many scenarios especially with non-expert users (Kahneman 2011; Herman et al. 2016; Reddy, Dragan, and Levine 2018; Gong and Zhang 2020). Having a biased belief about the dynamics could lead to biased (not just noisy) feedback for the agent's behavior, resulting in an inaccurate estimation of the human's reward function, sub-optimal behavior, and potentially undesired behavior to the humans.

Consider a mobile robot that is tasked to deliver packages on campus. Even though the robot is fully water-proof, when it is raining, a human user (due to the biased belief that the robot may be damaged in rain) may still prefer the robot to navigate under covered areas. In such cases, the robot may be misled to learn that navigating through covered areas generates more reward and choose to navigate through those areas even when it is sunny outside, leading to sub-optimal behaviors. To address such issues, the learning agents must remove the restrictive assumption that humans have a correct belief about the agent's dynamics, and take into consideration the human biased belief while learning a surrogate reward function. Thereby, the agent

is more like to recover the human's objectives and learn a policy with near-optimal performance.

Other than reward and policy learning under human biased belief about the domain dynamics, another problem setting where the human biases would impact the agent's decision-making is the need to behave explicably (i.e., being easily understood by the human and match the human's expectation) in a human-agent team. Consider a human is working with a coworker. It would be beneficial to the team if her behavior is explicable, especially not surprises or confuses her coworker. Similarly, when the agent accomplishes a task along with a human, choosing its optimal behavior without considering the human observer or collaborator could be seen as inexplicable. Such behaviors would increase the human cognitive load, undermine the human's trust of the agents and hurt the teaming performance to some extent. (Gunning 2017; Chakraborti, Kambhampati, et al. 2017; Zhang et al. 2016, 2017).

Consider an automated factory where a robotic agent is assisting a human co-worker to fetch various parts in an assembly task. The human would maintain expectations of the robot in terms of the objects the robot could handle. When the human has a biased belief about the agent's domain dynamics, the model difference would result in the agent's behaviors does not match the human's expectation. For example, when the human underestimates the robot's capability of handling high-value but delicate objects, she may intentionally choose to fetch an object by herself that would be more efficient for the robot, or try to stop the robot midway. If the robot still insists in fetching the delicate objects following its own optimal policy. Such inexplicable behavior would make the human lower her trust on it and consequently hurt the team performance. Therefore, to be a good team player, the agent is required to respect human expectation of its behaviors while accomplishing the tasks.

Learning the hidden expectations requires support from humans. Existing work on IRL (Ng and Russell 2000; Abbeel and Ng 2004; Ramachandran and Amir 2007; Ziebart et al. 2008; Ziebart, Bagnell, and Dey 2010) can be used to learn the human expectation using observed expert demonstrations, and reward learning methods (Daniel et al. 2014; Sadigh et al. 2017; Erdem et al. 2020) performs learning tasks from human feedback to learn the expectation. However, as we discussed above, they all implicitly assume that the human maintains an accurate understanding of the agent's domain dynamics and interprets any deviations from optimality as noise.

While noise introduces variations, bias determines the average of errors (Kahneman 2011). The misalignment between the human's expectation and an agent's chosen behaviors can be attributed to the human's biased belief about the domain dynamics. Similar to prior work, we assume that the human generates her expectation of the agent based on the belief about the dynamics and her objectives. Human biased belief can significantly impact her decisions and judgments. Hence, overlooking human belief in the agent's learning process could result in the agent's behaviors diverging from the human's expectations, while overweighting them could result in the agent deviating from its design purposes. Thus, addressing human biases is a fundamental problem of reconciliation.

A good team player is required to respect the others' expectations when appropriate, potentially at the cost of losing its own optimality. The challenge in developing such an ability for an intelligent agent lies in learning about the human's *hidden* expectations and planning to trade off optimality for meeting those expectations. Similar problems have been studied in classical planning (Zhang et al. 2017; Kulkarni et al. 2019; Chakraborti, Sreedharan, et al. 2017) but can hardly scale. Moreover, it remains unexplored in stochastic domains with continuous state and action spaces. Generally,

the agent is required to behave in an expected way and simultaneously be aware of the biased belief so as to trade off optimality with human preferences appropriately. Motivated by such scenarios, we emphasize the necessity of taking the human biased belief into account while policy learning in order to obtain efficiency as well as explicable behaviors.

1.2 Related Work

1.2.1 Learning from Human Feedback

Researchers have formulated the problem of inferring the human's intent and preferences as an IRL problem (Russell 1998) where the goal is to recover the human's preferences as a reward function. IRL is often solved using various optimization techniques with expert demonstrations as the input (Ng and Russell 2000; Abbeel and Ng 2004; Ziebart et al. 2008; Boularias, Kober, and Peters 2011; Ramachandran and Amir 2007). However, expert demonstrations (with or without noise) are often difficult to obtain in real-world tasks. More recently, researchers start focusing on learning with non-expert feedback on the queries of the intelligent agent's behaviors, often in the forms of ratings (Daniel et al. 2014), comparisons (Dorsa Sadigh, Sastry, and Seshia 2017), or critiques (Cui and Niekum 2018; Zhang and Dragan 2019).

Furthermore, Preference-based Reinforcement Learning (PbRL) has been successful in various tasks (Wirth and Fürnkranz 2013c). The preferences can be used to learn a policy directly via estimating a distribution of parametric policy space (Wilson, Fern, and Tadepalli 2012), or to compare and rank policies (Busa-Fekete et al. 2013, 2014). Moreover, a preference model can be learned, then it is used to get a ranking for actions

given a state, then derive a policy (Hüllermeier et al. 2008; Wirth and Fürnkranz 2013a, 2013b). However, these approaches suffer from feedback efficiency issues such that it needs many human preference data to achieve near-optimal policy. Another line of PbRL is to learn a surrogate reward function which can be used to optimize the policy. Christiano et al.(Christiano et al. 2017) models the reward function using deep neural networks which scale it to much complex tasks. PEBBLE (Lee, Smith, and Abbeel 2021) is proposed recently to improve the sample- and feedback- efficiency by unsupervised pretraining and off policy learning with reward relabeling. These prior work, however, rely on an implicit assumption that the non-expert user maintains a correct understanding of the agent's domain dynamics. When the user is biased, especially under non-expert settings, it may lead the agent to learning a wrong reward function and result in undesired behavior.

1.2.2 Model Asymmetry and Reconciliation in A Human-Agent Team

Researchers have been investigating the model differences between human and agent in the form of domain dynamics (Zhang et al. 2017; Chakraborti, Kambhampati, et al. 2017; Chakraborti et al. 2019; Reddy, Dragan, and Levine 2018; Gong and Zhang 2020). Formulated in the form of classical planning setting, Chakraborti et al. (Chakraborti, Sreedharan, et al. 2017; Kulkarni et al. 2019) leverage the differences between the domain models to generate explanations to the human. But, in these work, the human belief model is assumed to be given a prior which is impractical for the real world problems. Reddy et al. (Reddy, Dragan, and Levine 2018) manages to learn the human belief model in the presence of reward function using inverse soft

Q-learning method. Then the learned belief model is used to assist human decision making.

The problem of generating communicative actions or behaviors has been well studied as a subarea in explainable AI, where explicable planning is a representative method (Zhang et al. 2017; Kulkarni et al. 2019; Gong and Zhang 2018; Zakershahrak et al. 2018). The key characterization of explicable planning method revolves around the idea of model reconciliation where an agent makes decisions based on two models instead of one (Chakraborti, Sreedharan, et al. 2017; Chakraborti, Kambhampati, et al. 2017; Chakraborti et al. 2019). Zhang et al. (Zhang et al. 2017) formulated the problem as a learning and planning problem, where the human's expectation of the agent's behavior is learned through a labeling process, which captures the human's belief about the agent's dynamics. A metric for explicability is defined based on the learned labeling schema and then used to regularize the planning process to synthesize explicable plans. Kulkarni et al. (Kulkarni et al. 2019) considered it directly as a distance learning problem (Chakraborti, Sreedharan, et al. 2017) and generated explicable plans by minimizing an explicability distance between plans from the two models. A strong assumption was made that the human's model was provided a priori. These prior methods addressed the problem in a classical planning setting (e.g., PDDL) under deterministic domains, which is not suitable for stochastic environments with continuous state and action spaces. Furthermore, they considered the biased belief about dynamics only.

1.3 Dissertation Outline

The rest of the dissertation starts with a study in Chapter 2 on reward learning under human biased belief about the domain dynamics (Gong and Zhang 2020). We introduce an approach that estimates the human reward function while taking into account the human's belief about the agent's dynamics in the learning process. It demonstrates that the proposed methods can successfully recover the true reward function while a wrong reward functions are learned with baseline methods which have no concern about the human biased belief. It then followed by a work that aims to neutralize the effects of biased belief in preference-based reinforcement learning (Gong and Zhang 2022b) in Chapter 3. We focus on learning the agent policy from human feedback and we propose to learn a reward function and belief model together from the human feedback. We demonstrate that even if we may not learn a perfect reward function, we would find a reward function that describes what the human really wants the agent to behave and obtains improved performance in terms of trajectories returns on ground truth reward function, while the prior PbRL method shows poor performance. In Chapter 4, we will introduce the problem of explicable policy search (Gong and Zhang 2022a) and propose a solution that learns a surrogate reward function that encodes the information of the human's reward and belief about dynamics, and it can be easily integrated into policy optimization process to generate explicable and efficient behaviors. Finally, the dissertation is concluded in Chapter 5 with discussions on future work.

Chapter 2

GENERALIZED REWARD LEARNING WITH BIASED BELIEFS ABOUT DOMAIN DYNAMICS

2.1 Introduction

With the rapid advancement in AI and robotics, intelligent agents begin to play an important role in our lives in many different areas. Intelligent agent will soon be expected to not only achieve tasks alone, but also engage in tasks that require close collaboration with their human teammates. In such situations, the ability of the agent to understand the human's intent and preferences becomes a determinant for achieving effective human-agent teaming. The problem of inferring human's intent and preferences has been studied extensively before. Some researchers (Ng and Russell 2000) formulated this problem as an Inverse Reinforcement Learning (IRL) problem (Russell 1998). The reward function is recovered from optimal policies or behaviors demonstrated by human experts. Such expert demonstrations, however, are often difficult to obtain in real-world tasks. To address this problem, learning methods based on non-expert user ratings of the agent's behaviors (Daniel et al. 2014; Dorsa Sadigh, Sastry, and Seshia 2017; Cui and Niekum 2018) are developed. A common assumption made implicitly in all these prior works is that the human always maintains a correct understanding of the agent's domain dynamics. This, however, may not be the case in many scenarios especially with non-expert users. Having a biased belief about the agent could lead to biased (not just noisy) ratings for the agent's behaviors, resulting in an inaccurate estimation of the human's reward function.

Consider a robot vacuum cleaner that is tasked to clean the floors in a house. Suppose that the robot vacuum is designed to clean most floor types except for hardwood since it is too slippery for the robot to grasp onto (so it may be stuck in a room with a hardwood floor once entered). Consider a user who is asked to rate the robot's behaviors. Given a set of trajectories of the robot cleaner (with most of the areas covered except for the living room with a hardwood floor), the robot may get low ratings even though it should have received high ratings had the user known about the robot's capabilities (which are expressed in terms of domain dynamics). On the other hand, the robot may receive high ratings (even though it should not have) when it stays (stuck) in the living room but somehow manages to clean it (albeit much less efficiently), if the user had the belief that the robot was designed to clean only one room at a time.

In this work, we remove the restrictive assumption that humans have a correct belief about the agent's domain dynamics. Our goal is to recover the true reward function under biased beliefs. We refer to this problem as Generalized Reward Learning (GRL) and propose a method called *Generalized Reward Learning with biased beliefs about domain dynamics (GeReL)* that infers the latent variables governing both the reward function and human's belief together in a Bayesian setting based on human ratings of the agent's behaviors. Due to the complex forms of the posteriors, the problem is formulated in a variational inference framework (Jordan et al. 1999; Bishop 2006). The variational posterior distribution of the latent variables for estimating the true posterior is optimized using a black-box optimization method (Ranganath, Gerrish, and Blei 2014). To reduce the variance of Monte Carlo estimates of the variational gradients, we factorize the updating rules according to the independence of the latent variables and apply control variate to make the optimization converge

faster. By inferring the reward function and the human's belief about the agent simultaneously in this way, our learning method is able to recover the true human preferences while at the same time maintain an estimate of the human's biased belief. As such, our method addresses a key limitation of the existing methods and hence has broad impacts on improving the applicability and safety of robotic systems that work closely with humans.

To evaluate our method, we perform experiments in a simulated navigation domain and with a user study in the Coffee Robot domain (Boutilier, Dearden, and Goldszmidt 2000; Sigaud and Buffet 2013) where biases are introduced by varying the robot's appearances. We compare GeReL with a variant of Simultaneous Estimation of Rewards and Dynamics (SERD) (Herman et al. 2016), Maximum Entropy IRL (MaxEnt-IRL) (Ziebart et al. 2008), and another baseline approach that uses our inference method but maintains the same assumption as in MaxEnt-IRL (that the human's understanding of the agent is correct). In the latter two methods, the true domain dynamics is used and held fixed during learning. Results show that GeReL can better recover the true reward function under such biased beliefs when compared to these other methods. Furthermore, when biases are present, the learned preferences could be completely opposite to the ground truth, suggesting that such a method is indeed valuable for addressing biases in robotic applications.

2.2 Our Approach

The workflow of GeReL is presented in Figure 3. The intelligent agent will first randomly generate a set of demonstrations for querying the human for ratings. Then, the ratings of the demonstrations will be used to infer both the human's reward

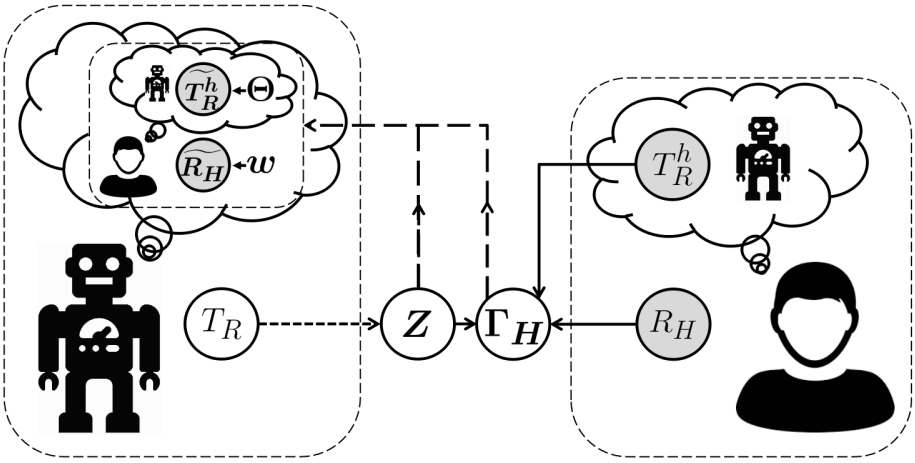


Figure 3. Workflow of GeReL. Using the robot's true transition model, the robot randomly generates a set of demonstrations which are evaluated by the human. The human is assumed to provide his rating for each instance according to the reward function and his belief about the robot's domain dynamics. The ratings will be used to update the estimated reward function and human's understanding of the robot. Gray circles denote the latent variables while the observed are in white.

function and his belief. The system terminates when it meets the convergence criterion. Similar to prior work on reward learning, we assume that the human is to always maximize the rewards (Ng and Russell 2000; Abbeel and Ng 2004), so that his ratings can be estimated given the reward function and his belief of the domain dynamics.

2.2.1 Problem Formulation

More specifically, given an agent's demonstration ζ, we assume that the human would rate it according to two factors, the reward function R_H and his belief about the agent's domain dynamics T_R^h. When the human's belief is different from the true agent's domain dynamics, the rating may be biased and could then lead to a wrong

interpretation of the human's preference. This setting introduces the *Generalized Reward Learning (GRL)* problem as follows:

Given:

- Agent's demonstrations $\boldsymbol{Z}$;
- Human's ratings $\boldsymbol{\Gamma_H}$ for each instance in $\boldsymbol{Z}$.

To determine:

- Human's true reward function R_H;
- Human's belief T_R^h about agent's domain dynamics.

To solve this problem, we formulate the environment as a Markov Decision Processes (MDP). An MDP is defined by a tuple (S, A, R, T, λ) where S is a finite set of states, A is a finite set of actions, and $R : S \mapsto \mathbb{R}$ is the reward function that maps each state to a utility value. $T : S \times A \times S \mapsto [0, 1]$ is the transition function that specifies the probability of transitioning to the next state when you take an action in the current state. λ is the discount factor that determines how the agent favors current rewards over future rewards.

Similar to prior work on reward learning (Ng and Russell 2000; Abbeel and Ng 2004), we formulate the reward function R_H for a state s as follows:

$$R_H(s) = \boldsymbol{w} \cdot \Phi(s)$$

where $\Phi = [\phi_0, \phi_1, \ldots, \phi_k]^T$ denotes a set of predefined features for states and $\boldsymbol{w} = [w_0, w_1, \ldots, w_k]^T$ denotes a set of weights for the features. The agent's domain dynamics (i.e., the true domain dynamics) is captured by a transition function and assumed to be given. Likewise, the human's belief about the agent's domain dynamics is modeled also as a transition function T_R^h, which is hidden. T_R^h is assumed to follow a set of probability distributions $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_{|S|\times|A|}]$ where $\boldsymbol{\theta}_i = p(s'|s, a)$ is a distribution for a fixed s and a. These distributions capture the human's prior belief about the agent.

To rate a agent's behavior ζ, we assume that the human will first generate his expectation of the agent's behavior as an optimal policy generated using his reward function R_H and belief about the agent T_R^h. Then the behavior of the agent is compared with the optimal policy to generate a rating γ_H. Hence, our learning task in this work becomes to learn the weights $\boldsymbol{w}$ and transition probability distributions $\boldsymbol{\Theta}$.

2.2.2 Methodology

The inference problem above is often solved by optimizing the posterior probability with respect to the latent variables. However, due to the complex forms of the posteriors, we formulate the problem in a variational inference framework (Jordan et al. 1999; Bishop 2006). Our goal is to approximate the posterior distribution $p(\boldsymbol{w}, \boldsymbol{\Theta}|\boldsymbol{\Gamma_H}, \boldsymbol{Z})$, where $\boldsymbol{\Gamma_H}, \boldsymbol{Z}$ are the observations and $\boldsymbol{w}$, $\boldsymbol{\Theta}$ the latent variables.

We assume that $\boldsymbol{w}$ follows a multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. For simplicity, we assume that $\boldsymbol{\Sigma}$ is given a priori. For $\boldsymbol{\Theta}$, we need to select a prior for each $\boldsymbol{\theta}_i$ as a probability distribution. We assume that each $\boldsymbol{\theta}_i$ follows a Dirichlet distribution $\textsc{Dir}(\boldsymbol{\alpha}_i)$, which encodes a distribution over distributions. Let $\boldsymbol{\mathcal{A}} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_{|S|\times|A|}]$, and thereby, $\boldsymbol{\mu}$ and $\boldsymbol{\mathcal{A}}$ are the parameters we need to learn. As a result, our variational posterior distribution becomes $q(\boldsymbol{w}, \boldsymbol{\Theta}|\boldsymbol{\mu}, \boldsymbol{\mathcal{A}})$, which is the posterior of the latent variables that correspond to the reward function and human's belief about the agent's domain dynamics governed by $\boldsymbol{\mu}$ and $\boldsymbol{\mathcal{A}}$. It thus transforms the problem of inferring R_H and T_R^h into a problem of finding $\boldsymbol{\mu}$ and $\boldsymbol{\mathcal{A}}$ to make $q(\boldsymbol{w}, \boldsymbol{\Theta}|\boldsymbol{\mu}, \boldsymbol{\mathcal{A}})$ to be close to $p(\boldsymbol{w}, \boldsymbol{\Theta}|\boldsymbol{\Gamma_H}, \boldsymbol{Z})$.

As a variational inference problem, we optimize the Evidence Lower BOund

(ELBO):

$$\mathcal{L}(q) = \mathbb{E}_{q(\boldsymbol{w},\boldsymbol{\Theta})}\left[\log p(\boldsymbol{\Gamma}_{\boldsymbol{H}}, \boldsymbol{Z}, \boldsymbol{w}, \boldsymbol{\Theta}) - \log q(\boldsymbol{w}, \boldsymbol{\Theta})\right]$$

where $p(\boldsymbol{\Gamma}_{\boldsymbol{H}}, \boldsymbol{Z}, \boldsymbol{w}, \boldsymbol{\Theta})$ is the joint probability of the observations $\boldsymbol{\Gamma}_{\boldsymbol{H}}, \boldsymbol{Z}$ and latent variables $\boldsymbol{w}, \boldsymbol{\Theta}$. In order to make it computable in our task, we apply black-box variational inference (Ranganath, Gerrish, and Blei 2014) to maximize ELBO via stochastic optimization:

$$\langle \boldsymbol{\mu}, \boldsymbol{\mathcal{A}} \rangle = \langle \boldsymbol{\mu}, \boldsymbol{\mathcal{A}} \rangle + \rho \cdot \nabla_{\langle \boldsymbol{\mu}, \boldsymbol{\mathcal{A}} \rangle} \mathcal{L}(q)$$

where the learning rate ρ follows the Robbins-Monro rules (Robbins and Monro 1951). We compute the gradient of ELBO with respect to the free parameters $\boldsymbol{\mu}$ and $\boldsymbol{\mathcal{A}}$ and $\nabla_{\langle \boldsymbol{\mu}, \boldsymbol{\mathcal{A}} \rangle} \mathcal{L}(q)$ is derived as follows:

$$\begin{aligned}\nabla_{\langle \boldsymbol{\mu}, \boldsymbol{\mathcal{A}} \rangle} \mathcal{L}(q) = \mathbb{E}_{q(\boldsymbol{w},\boldsymbol{\Theta})}[\nabla_{\langle \boldsymbol{\mu}, \boldsymbol{\mathcal{A}} \rangle} \log q(\boldsymbol{w}, \boldsymbol{\Theta} | \boldsymbol{\mu}, \boldsymbol{\mathcal{A}}) \\ \cdot (\log p(\boldsymbol{\Gamma}_{\boldsymbol{H}}, \boldsymbol{Z}, \boldsymbol{w}, \boldsymbol{\Theta}) - \log q(\boldsymbol{w}, \boldsymbol{\Theta} | \boldsymbol{\mu}, \boldsymbol{\mathcal{A}}))] \quad (2.1)\end{aligned}$$

From Equation (2.1), we can see that the gradient of ELBO is the expectation of the multiplication of the *score function* (Hinkley and Cox 1979) (i.e., $\nabla_{\langle \boldsymbol{\mu}, \boldsymbol{\mathcal{A}} \rangle} \log q(\boldsymbol{w}, \boldsymbol{\Theta} | \boldsymbol{\mu}, \boldsymbol{\mathcal{A}})$) and *instantaneous ELBO* (i.e., $\log p(\boldsymbol{\Gamma}_{\boldsymbol{H}}, \boldsymbol{Z}, \boldsymbol{w}, \boldsymbol{\Theta}) - \log q(\boldsymbol{w}, \boldsymbol{\Theta} | \boldsymbol{\mu}, \boldsymbol{\mathcal{A}})$) with respect to our variational posterior distribution. The detailed derivation of $\nabla_{\langle \boldsymbol{\mu}, \boldsymbol{\mathcal{A}} \rangle} \mathcal{L}(q)$ is presented in (Ranganath, Gerrish, and Blei 2014).

The form of $\nabla_{\langle \boldsymbol{\mu}, \boldsymbol{\mathcal{A}} \rangle} \mathcal{L}(q)$ is not directly computable. Given that $\boldsymbol{\mu}$ and $\boldsymbol{\mathcal{A}}$ are independent parameters in our setting, we compute $\nabla_{\boldsymbol{\mu}} \mathcal{L}(q)$ and $\nabla_{\boldsymbol{\mathcal{A}}} \mathcal{L}(q)$ respectively and update them separately:

$$\boldsymbol{\mu} = \boldsymbol{\mu} + \rho_{\boldsymbol{\mu}} \cdot \nabla_{\boldsymbol{\mu}} \mathcal{L}(q)$$

$$\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{A}} + \rho_{\boldsymbol{\mathcal{A}}} \cdot \nabla_{\boldsymbol{\mathcal{A}}} \mathcal{L}(q)$$

This also allows us to apply the mean-field assumption that gives the following factorization:

$$q(\boldsymbol{w}, \boldsymbol{\Theta}|\boldsymbol{\mu}, \boldsymbol{\mathcal{A}}) = q(\boldsymbol{w}|\boldsymbol{\mu}) \cdot q(\boldsymbol{\Theta}|\boldsymbol{\mathcal{A}})$$

Then we can rewrite the gradient of ELBO as follows:

$$\nabla_{\langle \boldsymbol{\mu}, \boldsymbol{\mathcal{A}} \rangle} \mathcal{L}(q) = \mathbb{E}_{q(\boldsymbol{w})} \mathbb{E}_{q(\boldsymbol{\Theta})} \left[\nabla_{\langle \boldsymbol{\mu}, \boldsymbol{\mathcal{A}} \rangle} (\log q(\boldsymbol{w}|\boldsymbol{\mu}) + \log q(\boldsymbol{\Theta}|\boldsymbol{\mathcal{A}})) \right.$$
$$\left. \cdot (\log p(\boldsymbol{\Gamma_H}, \boldsymbol{Z}, \boldsymbol{w}, \boldsymbol{\Theta}) - \log q(\boldsymbol{w}|\boldsymbol{\mu}) - \log q(\boldsymbol{\Theta}|\boldsymbol{\mathcal{A}})) \right]$$

Take $q(\boldsymbol{w}|\boldsymbol{\mu})$ as an example, following the derivations in (Ranganath, Gerrish, and Blei 2014), the gradient of ELBO with respect to $\boldsymbol{\mu}$ becomes:

$$\nabla_{\boldsymbol{\mu}} \mathcal{L}(q) = \mathbb{E}_{q(\boldsymbol{w})} \left[\nabla_{\boldsymbol{\mu}} (\log q(\boldsymbol{w}|\boldsymbol{\mu})) \cdot \mathbb{E}_{q(\boldsymbol{\Theta})} \left[\log p(\boldsymbol{\Gamma_H}, \boldsymbol{Z}, \boldsymbol{w}, \boldsymbol{\Theta}) - \log q(\boldsymbol{w}|\boldsymbol{\mu}) - \log q(\boldsymbol{\Theta}|\boldsymbol{\mathcal{A}}) \right] \right]$$

Note that the first term $\mathbb{E}_{q(\boldsymbol{w})} \left[\nabla_{\boldsymbol{\mu}} \left(\log q(\boldsymbol{w}|\boldsymbol{\mu}) \right] = 0$ (Ranganath, Gerrish, and Blei 2014). Hence the last term in the *instantaneous ELBO* can be considered as a constant with respect to $q(\boldsymbol{w})$ and canceled out. $\nabla_{\boldsymbol{\mu}} \mathcal{L}(q)$ then becomes:

$$\nabla_{\boldsymbol{\mu}} \mathcal{L}(q) = \mathbb{E}_{q(\boldsymbol{w})} \left[\nabla_{\boldsymbol{\mu}} (\log q(\boldsymbol{w}|\boldsymbol{\mu})) \cdot (\mathbb{E}_{q(\boldsymbol{\Theta})} \left[\log p(\boldsymbol{\Gamma_H}, \boldsymbol{Z}, \boldsymbol{w}, \boldsymbol{\Theta}) \right] - \log q(\boldsymbol{w}|\boldsymbol{\mu})) \right] \quad (2.2)$$

Different from (Ranganath, Gerrish, and Blei 2014), in our problem, the expectation of the log joint probability $\log p(\boldsymbol{\Gamma_H}, \boldsymbol{Z}, \boldsymbol{w}, \boldsymbol{\Theta})$ cannot be canceled out since $\boldsymbol{w}$ and $\boldsymbol{\Theta}$ happen to be in the Markov blanket of each other. Based on the relationship among these variables as shown in Figure 3, the log probability can be factorized as follows:

$$\log p(\boldsymbol{\Gamma_H}, \boldsymbol{Z}, \boldsymbol{w}, \boldsymbol{\Theta}) = \log p(\boldsymbol{\Gamma_H}|\boldsymbol{Z}, \boldsymbol{w}, \boldsymbol{\Theta}) + \log p(\boldsymbol{w}) + \log p(\boldsymbol{\Theta}) + \log p(\boldsymbol{Z}) \quad (2.3)$$

Putting Equation (2.3) back into Equation (2.2), we obtain:

$$\nabla_{\boldsymbol{\mu}} \mathcal{L}(q) = \mathbb{E}_{q(\boldsymbol{w})} \left[\nabla_{\boldsymbol{\mu}} (\log q(\boldsymbol{w}|\boldsymbol{\mu})) \right.$$
$$\left. \cdot (\mathbb{E}_{q(\boldsymbol{\Theta})} \left[\log p(\boldsymbol{\Gamma_H}|\boldsymbol{Z}, \boldsymbol{w}, \boldsymbol{\Theta}) \right] + \log p(\boldsymbol{w}) - \log q(\boldsymbol{w}|\boldsymbol{\mu})) \right] \quad (2.4)$$

where the expectation of the terms $\log p(\boldsymbol{\Theta})$ and $\log p(\boldsymbol{Z})$ with respect to $q(\boldsymbol{\Theta})$ are constants and can be canceled out since $\mathbb{E}_{q(\boldsymbol{w})}[\nabla_{\boldsymbol{\mu}}(\log q(\boldsymbol{w}|\boldsymbol{\mu})] = 0$. Now we have obtained the gradient of ELBO with respect to the latent variable $\boldsymbol{\mu}$ as presented in Equation (2.4). Similarly, the gradient of ELBO with respect to each $\boldsymbol{\alpha}_i \in \boldsymbol{\mathcal{A}}$ is as follows:

$$\nabla_{\boldsymbol{\alpha}_i}\mathcal{L}(q) = \mathbb{E}_{q(\boldsymbol{\theta}_i)}\left[\nabla_{\boldsymbol{\alpha}_i}(\log q(\boldsymbol{\theta}_i|\boldsymbol{\alpha}_i)) \cdot (\mathbb{E}_{q(\boldsymbol{w})}\left[\log p(\boldsymbol{\Gamma_H}|\boldsymbol{Z}, \boldsymbol{w}, \boldsymbol{\Theta})\right] + \log p(\boldsymbol{\theta}_i) - \log q(\boldsymbol{\theta}_i|\boldsymbol{\alpha}_i))\right]$$

Both $p(\boldsymbol{w})$ and $p(\boldsymbol{\theta_i})$ are priors, which are assumed to follow a multivariate Gaussian distribution and a Dirichlet distribution respectively.

In the equations above, $\log p(\boldsymbol{\Gamma_H}|\boldsymbol{Z}, \boldsymbol{w}, \boldsymbol{\Theta}) = \sum \log p(\gamma_H|\zeta, \boldsymbol{w}, \boldsymbol{\Theta})$ since the demonstrations and ratings are conditionally independent from each other. $p(\gamma_H|\zeta, \boldsymbol{w}, \boldsymbol{\Theta})$ indicates how likely the human would give a rating γ_H for the demonstration ζ given the parameters for the reward function and human's belief about the agent. We assume a Gaussian distribution $\mathcal{N}(\gamma_H|\widetilde{\gamma_H}, \Sigma_{\gamma_H})$ where $\widetilde{\gamma_H}$ is the estimated mean of the human's ratings given $\boldsymbol{w}$ and $\boldsymbol{\Theta}$, and Σ_{γ_H} is assumed to be given to simplify the discussion. As discussed earlier, the estimated mean rating of a demonstration is assumed to depend on two factors, the reward function and the human's belief about the agent's domain dynamics. They together determine the human's expectation of the agent's behavior, which corresponds to the optimal policy for the agent in the human's mind. In this work, we assume that the rating is proportional to the geometric mean of the human's softmax policy applied to the demonstration. Moreover, we define $\gamma_{\max}$ to be a constant that represents the highest rating that may be given. Following our discussion, the estimated human's rating can be computed for a demonstration $\zeta = \{(s_1, a_1), (s_2, a_2) \ldots (s_n, a_n)\}$ as:

$$\widetilde{\gamma_H} = \gamma_{\max} \cdot \left(\prod_{i=1}^{n} \widetilde{\pi}(a_i|s_i)\right)^{\frac{1}{n}}$$

where n is the length of the demonstration, and $\widetilde{\pi}$ is the estimated human's softmax policy computed using $\boldsymbol{w}$ and $\boldsymbol{\Theta}$.

Variance Reduction: The computation of $\nabla_{\boldsymbol{\mu}}\mathcal{L}(q)$ and $\nabla_{\boldsymbol{\alpha}_i}\mathcal{L}(q)$ above cannot be performed directly due to the intractability of computing the expectations. Hence, we approximate the gradients using sampling methods (Hastings 1970). With Monte Carlo samples, the gradients are estimated as follows:

$$\hat{\nabla}_{\boldsymbol{\mu}}\mathcal{L}(q) \triangleq \frac{1}{S}\sum_{s=1}^{S}\left[\nabla_{\boldsymbol{\mu}}\left(\log q(\boldsymbol{w}_s|\boldsymbol{\mu})\right)\cdot\left(\log p(\boldsymbol{\Gamma}_{\boldsymbol{H}}|\boldsymbol{Z},\boldsymbol{w}_s,\boldsymbol{\Theta}_s)+\log p(\boldsymbol{w}_s)-\log q(\boldsymbol{w}_s|\boldsymbol{\mu})\right)\right]$$

where S is the number of samples, and $\boldsymbol{w}_s \sim q(\boldsymbol{w})$, $\boldsymbol{\Theta}_s \sim q(\boldsymbol{\Theta})$.

These estimated gradients, however, may have a large variance which could hinder the convergence of our approach. Therefore, it is necessary to reduce the variance. (Ross 2002) introduced control variate that represents a family of functions with equivalent expectations. With control variate, we can instead compute the expectation of an alternative function which has a smaller variance. Let f be the function to be approximated, function $\hat{f}$ is defined as:

$$\hat{f} = f - a\cdot(g-\mathrm{E}[g])$$

where g serves as an auxiliary function that has a finite first moment. $\hat{f}$ can be proven to have smaller variances with an equivalent expectation, where the factor a is computed to minimize the variance (Ranganath, Gerrish, and Blei 2014) as $a = \frac{\mathrm{cov}(f,g)}{\mathrm{var}(g)}$. In this work, we select the expectation of the *score function* (i.e., $\mathbb{E}_{q(\boldsymbol{w})}\left[\nabla_{\boldsymbol{\mu}}\left(\log q(\boldsymbol{w}|\boldsymbol{\mu})\right)\right]$ and $\mathbb{E}_{q(\boldsymbol{\Theta})}\left[\nabla_{\boldsymbol{\alpha}_i}\left(\log q(\boldsymbol{\Theta}|\boldsymbol{\alpha}_i)\right)\right]$) to be g.

We present GeReL in Algorithm 1. Given the agent's demonstrations and the corresponding ratings, we leverage the human's ratings to update the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\mathcal{A}}$ of our variational posteriors $q(\boldsymbol{w})$ and $q(\boldsymbol{\Theta})$ via stochastic optimization. The gradients of the ELBO with respect to $\boldsymbol{\mu}$ and $\boldsymbol{\mathcal{A}}$ are approximated using Monte Carlo

Algorithm 1 Generalized Reward Learning with Biased Belief about Domain Dynamics (GeReL)

Input: the agent's demonstrations $\boldsymbol{Z}$, human's ratings $\boldsymbol{\Gamma_H}$, variational posteriors $q(\boldsymbol{w})$ and $q(\boldsymbol{\Theta})$, *MaxIter*
Output: $\boldsymbol{\mu}$ and $\boldsymbol{\mathcal{A}}$

Initialize: free parameters $\boldsymbol{\mu}$ and $\boldsymbol{\mathcal{A}}$ for $q(\boldsymbol{w})$ and $q(\boldsymbol{\Theta})$
Let $t = 1$.
while $t <$ *MaxIter* or convergence not met **do**
  Draw S samples from $q(\boldsymbol{w})$ and $q(\boldsymbol{\Theta})$
  for $s = 1$ to n **do**
    $\widetilde{\pi} \leftarrow$ the human's expected policy for the agent
    $\widetilde{\Gamma_H} \leftarrow$ estimated human's ratings for $\boldsymbol{Z}$ given $\widetilde{\pi}$
    Compute $f_{\boldsymbol{\mu}}$, $g_{\boldsymbol{\mu}}$, $f_{\boldsymbol{\alpha}_i}$, and $g_{\boldsymbol{\alpha}_i}$
  end for
  Compute $a_{\boldsymbol{\mu}}$ and $a_{\boldsymbol{\alpha}_i}$
  Approximate $\hat{\nabla}_{\boldsymbol{\mu}} L \triangleq \frac{1}{S}\sum_{s=1}^{S}[f_{\boldsymbol{\mu}} - a_{\boldsymbol{\mu}} g_{\boldsymbol{\mu}}]$ and $\hat{\nabla}_{\boldsymbol{\alpha}_i} L \triangleq \frac{1}{S}\sum_{s=1}^{S}[f_{\boldsymbol{\alpha}_i} - a_{\boldsymbol{\alpha}_i} g_{\boldsymbol{\alpha}_i}]$
  Compute learning rates $\rho_{\boldsymbol{\mu}}$ and $\rho_{\boldsymbol{\alpha}_i}$ with $\hat{\nabla}_{\boldsymbol{\mu}} L$, $\hat{\nabla}_{\boldsymbol{\alpha}_i} L$
  Update $\boldsymbol{\mu} = \boldsymbol{\mu} + \rho_{\boldsymbol{\mu}} \hat{\nabla}_{\boldsymbol{\mu}} L$ and $\boldsymbol{\alpha}_i = \boldsymbol{\alpha}_i + \rho_{\boldsymbol{\alpha}_i} \hat{\nabla}_{\boldsymbol{\alpha}_i} L$
end while
return $\boldsymbol{\mu}$ and $\boldsymbol{\mathcal{A}}$

sampling. Furthermore, we take advantage of control variate to reduce the variance of the gradient estimates. Lastly, the parameters, $\boldsymbol{\mu}$ and $\boldsymbol{\mathcal{A}}$, are updated in each iteration with an adapted learning rate based on AdaGrad (Duchi, Hazan, and Singer 2011). GeReL terminates when the convergence criterion is met.

2.3 Evaluation

To evaluate our approach, we conduct two sets of experiments in a simulated grid-world navigation domain and a Coffee Robot domain (Boutilier, Dearden, and Goldszmidt 2000; Sigaud and Buffet 2013) with a user study. The simulation will be focusing on validating our learning method under biased beliefs. The user study will serve two purposes, showing that 1) human users are easily biased in our problem

setting; 2) our algorithm learns the correct human preferences under such biases, while prior methods that ignore such biases would fail.

2.3.1 Simulated Navigation Domain

In the first experiment, we test the performance of GeReL in a grid-world navigation domain which contains $7 \times 7 = 49$ states. We set one reward state (i.e., location) that has a large positive weight (i.e. 5) and one penalty location with a large negative weight (i.e., -5). They are randomly located at corners of the grid-world. The agent starts at a random state and its goal is to maximize the rewards. There are four actions, {1 (Up), 2 (Down), 3 (Left), 4 (Right)}, which can transfer the agent from the current state to another state.

To test our algorithm, we simulate two types of biased human beliefs about the agent's domain dynamics. 1) *Reversed Up & Down* : the human believes that action 1 would take the agent down and action 2 would move it up instead. 2) *Rotated Belief*: human believes that the action 1 would move the agent left, the action 2 would move it right, the action 3 would move it up and the action 4 would move it down. The human's reward function for each state is defined as a weighted summation of an inverse distance metric to the reward and penalty states (i.e., the closer it is to the state, the more influence that state has on its reward). The demonstrations are randomly generated via the agent's true dynamic model. The human's ratings are simulated using the true human reward function and biased belief about the agent's domain dynamics following a Gaussian distribution.

We compare GeReL with two baseline methods that assume that the human maintains the correct belief about the agent's domain dynamics, namely MaxEnt-IRL

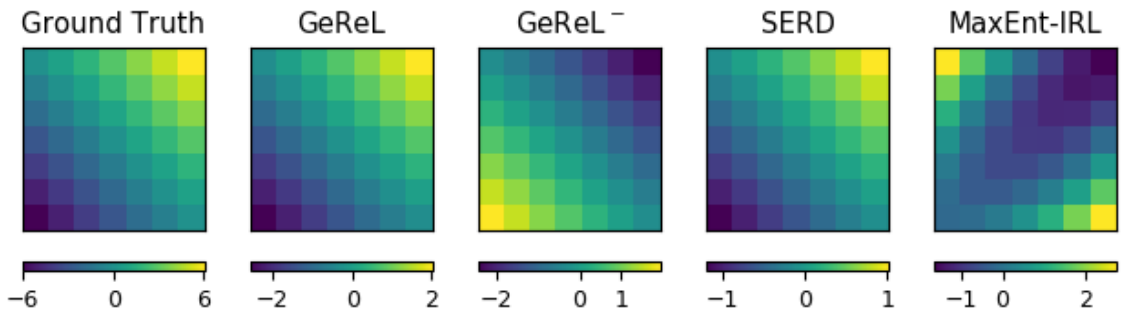


Figure 4. Rewards learned by different approaches.

(Ziebart et al. 2008) and GeReL$^-$, with the latter basically uses GeReL without updating the domain dynamics. In both baseline methods, the true agent's domain dynamics is used during learning. In addition, a variant of the Simultaneous Estimation of Rewards and Dynamics (SERD) algorithm (Herman et al. 2016) implemented that learns both the reward function and dynamics based on ratings (the original method does not apply to our problem setting) is used in the comparison, which relies on soft value iteration that requires the value functions to assume certain shapes to perform well. To obtain demonstrations for MaxEnt-IRL, we generate them based on the softmax policy of the human. All of the four methods are provided with the same amount of demonstrations. All the results are averaged over multiple runs.

Figure 5, 6, 7 show the results for the *Reversed Up & Down* setting. The result shows that GeReL can successfully recover the human's reward function and belief about the agent's domain dynamics while GeReL$^-$ and MaxEnt-IRL converge in the completely opposite direction since they do not consider that the human's belief could be biased. On the other hand, SERD converges in the right direction, but the learned values are farther from the ground truth than GeReL in all cases. This is due to the smoothing effect of soft value iteration. In addition, we compute the KL divergence of the softmax policy generated by the estimated reward function and

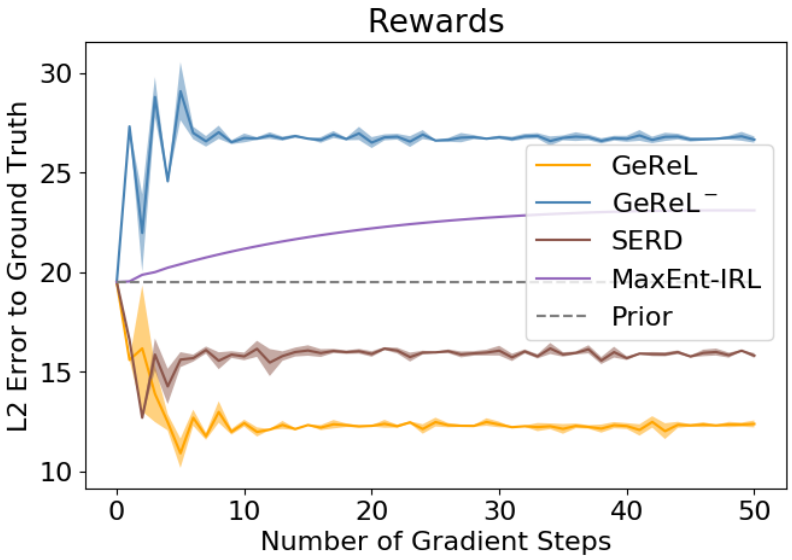


Figure 5. Comparison of the performance in terms of reward learning among GeReL, SERD, GeReL$^-$, MaxEnt-IRL with the prior also shown.

human's belief with that of the ground truth to examine how well we can estimate the human's expectation of the agent's behaviors. Similar trends are observed among all the methods.

The comparison of the rewards learned by these four methods with the ground truth is presented in Figure 4. Both GeReL and SERD converge to the correct pattern of rewards in terms of their relative magnitudes. SERD shows less sensitivity to the magnitudes since soft Bellman equation would lead to an entropy augmented reward function (Haarnoja et al. 2017). The adverse effect of learning from biased ratings is clear from the figure for GeReL$^-$ and MaxEnt-IRL, which both fail to recover the true preferences. The results for both settings are presented in Table 1, which show similar performances in between the two settings. It confirms that GeReL can effectively estimate the human's reward function under biased beliefs.

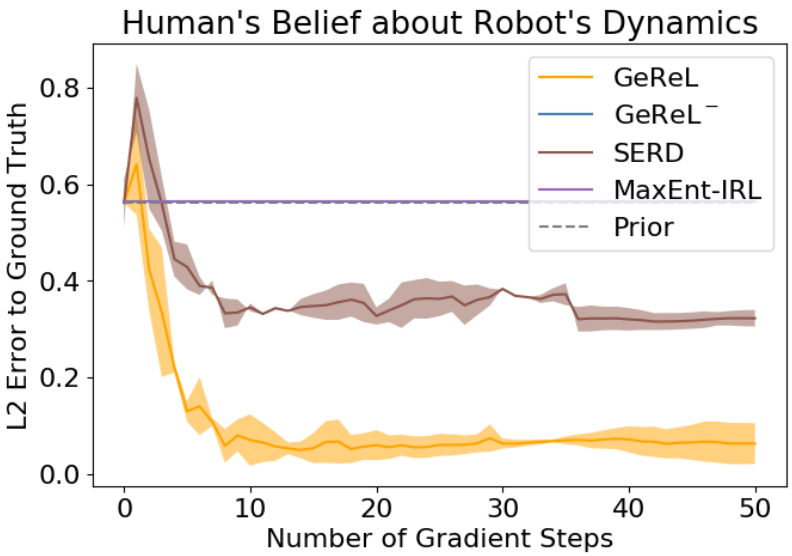


Figure 6. Comparison of the performance in terms of belief learning among GeReL, SERD, GeReL$^-$, MaxEnt-IRL with the prior also shown.

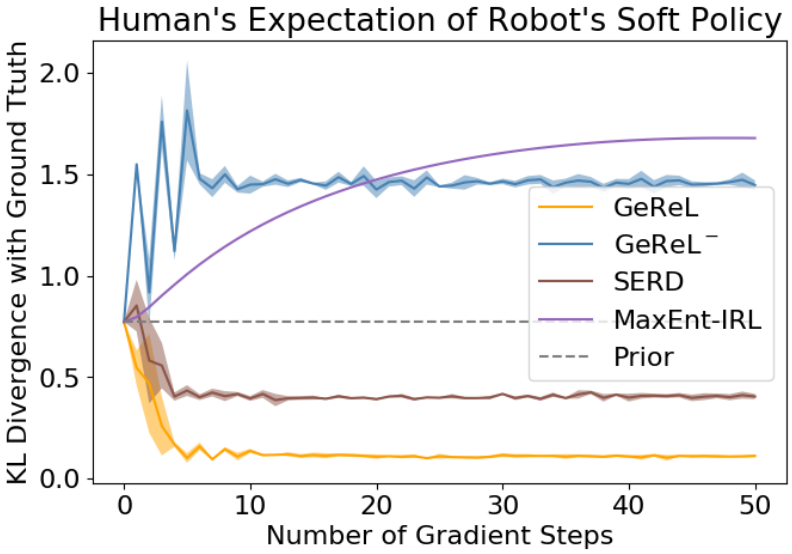


Figure 7. Comparison of the performance in terms of expectation recovery among GeReL, SERD, GeReL$^-$, MaxEnt-IRL with the prior also shown.

	$d(\boldsymbol{R})$	$d(\boldsymbol{\Theta})$	$d(\pi)$	$d(\boldsymbol{R})$	$d(\boldsymbol{\Theta})$	$d(\pi)$
	Reversed Up & Down			*Rotated Belief*		
GeReL	**12.17**	**0.06**	**0.11**	**12.61**	**0.08**	**0.23**
SERD	16.43	0.38	0.47	17.62	0.57	0.63
GeReL$^-$	26.72	0.56	1.46	23.96	0.91	1.43
MaxEnt-IRL	23.32	0.56	1.68	28.02	0.91	1.55

Table 1. Comparison of GeReL, SERD, GeReL$^-$, and MaxEnt-IRL for the two settings in our simulation with respect to the L_2 distance between the estimated values and the ground truth. The third column (i.e., $d(\pi)$) shows the KL divergence between the estimated human's expectation of the agent's softmax policy and that under the ground truth.

2.3.2 User Study with the Coffee Robot Domain

Besides the experiments in a simulated domain, we also conduct a user study. Through the study, we hope to demonstrate that humans can be easily biased in our problem setting, which may lead to biased ratings that could have led to a wrong interpretation of the human preference. In such cases, we will show that GeReL can accurately identify the situation. We apply the Coffee Robot domain (Boutilier, Dearden, and Goldszmidt 2000; Sigaud and Buffet 2013) in this user study, which is illustrated in Figure 8. This is a typical factored MDP domain described by 6 binary features which represent whether it is raining, whether the robot has a coffee, etc. The task of the robot is to buy a cup of coffee from a cafe and deliver it to a person in his office. When it is raining, the robot could choose to either operate in the rain or use an umbrella to stay dry. However, using an umbrella while holding the coffee cup may cause the coffee to spill.

To create a situation where biases may be present, we design two experimental settings with two different types of robots: a mobile robot and a humanoid, as seen in Figure 8. We anticipate that the appearance would introduce human biases (Haring

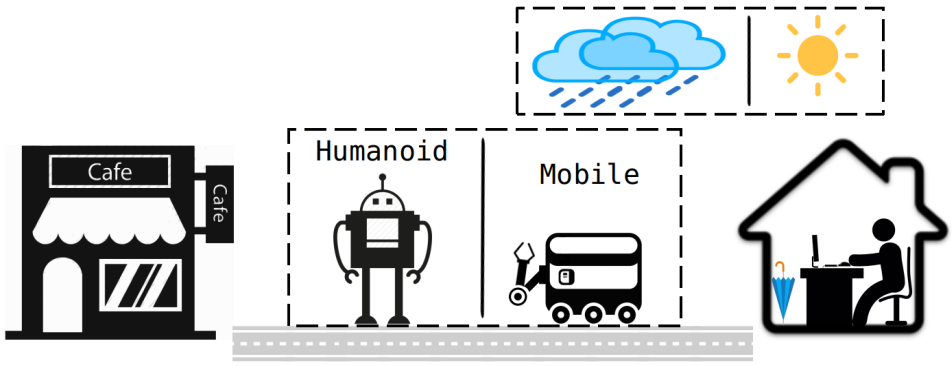


Figure 8. The Coffee Robot domain. The weather could be rainy or sunny, and the robot may choose to use an umbrella or operate in the rain. We use two types of robots (humanoid vs. mobile) to perform the same set of demonstrations that cover the various situations that may occur.

et al. 2018) in terms of their capabilities of handling the task. To reduce the effects that the human subject would improve their understanding over time, we generate only 8 demonstrations for each robot that include various scenarios that may occur, such as for a sunny or rainy day, for whether or not the robot takes the umbrella, and for whether or not the robot spills the coffee. *The ground truth for the domain dynamics is set such that the humanoid is less likely to spill the coffee while using the umbrella than the mobile robot.*

We publish the experiments on Amazon Mechanical Turk (MTurk), as shown in Figure 9, 10, 11. To remove invalid responses, we insert a sanity check demonstration with random actions, which should have received the lowest rating. We recruited 20 participants for each setting. After removing those that failed the sanity check or with very short response time ($<$ 3 min), we obtained 12 valid responses for each setting with ages ranging from 23 to 61 (the ratio of males to females is $2:1$). Each

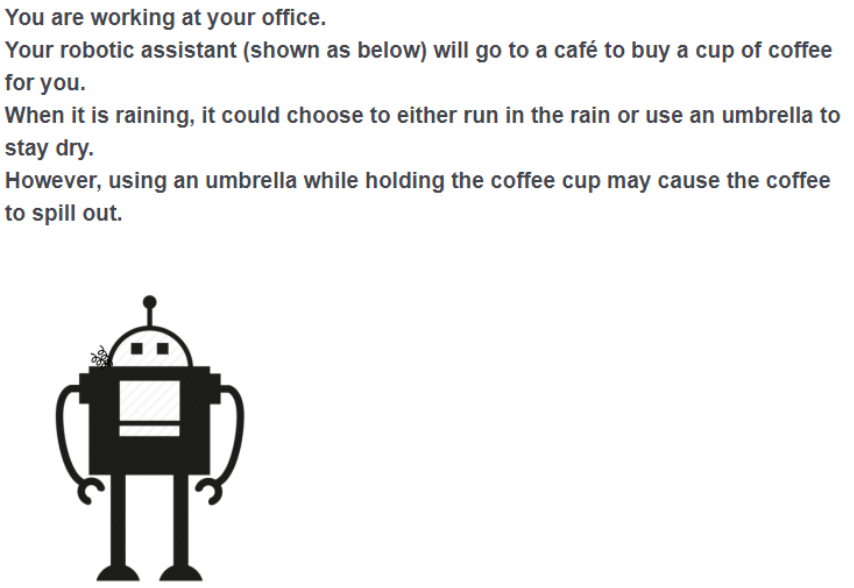


Figure 9. The introduction of the tasks on MTurk. We present the appearance of one of the two types of robots (humanoid) to the participants.

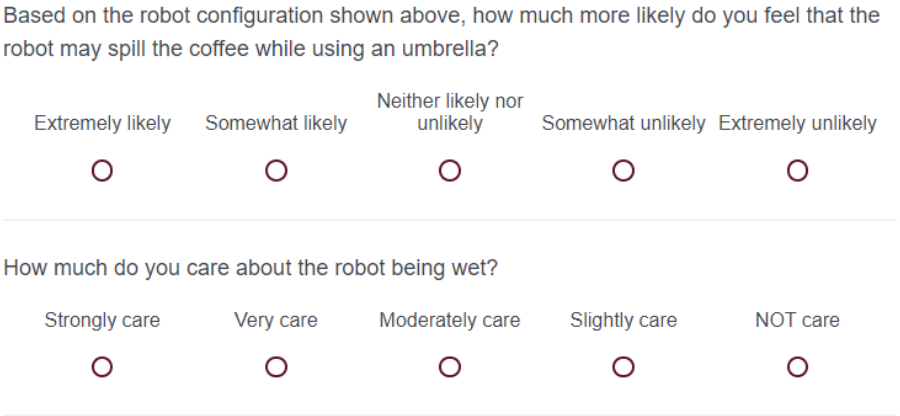


Figure 10. Questions for eliciting the paticipants' belief and rewards of the domain and task.

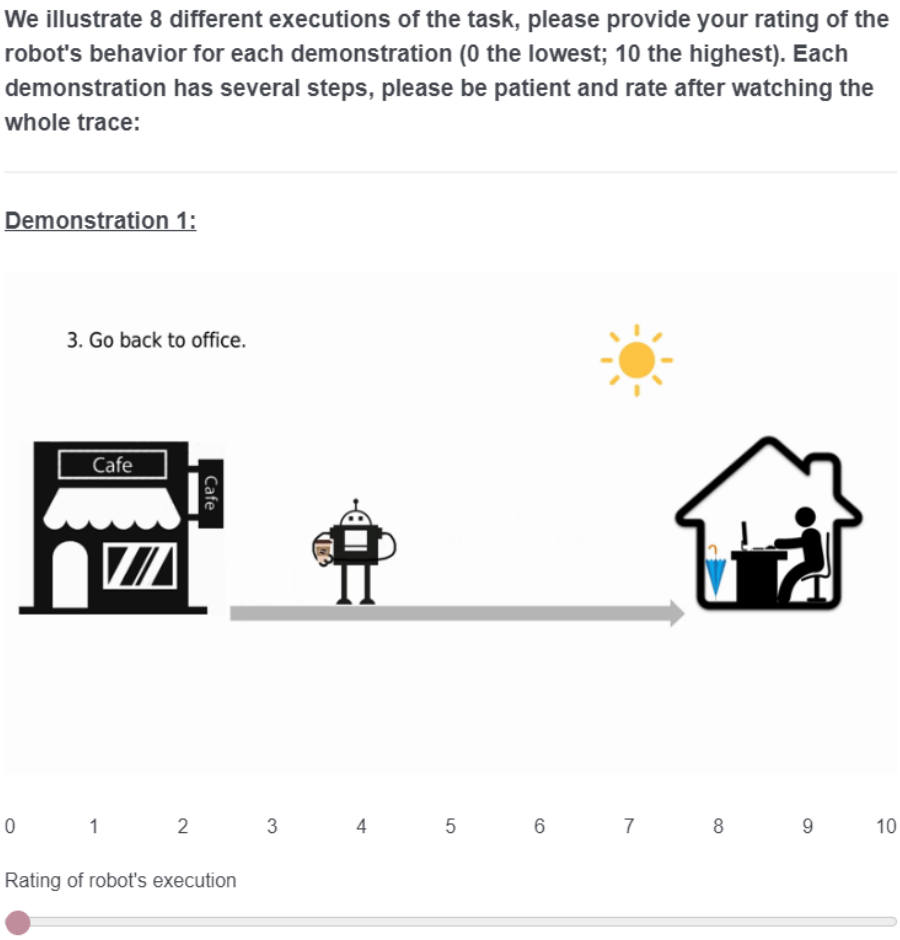


Figure 11. One demonstration of the coffee robot is illustrated to the participants. They need to rate it using the slider below.

participant is provided with instructions about the domain at the beginning. To avoid the influence from viewing the demonstrations, immediately after the instructions, we ask the participants two questions as follows:

- **Q1:** *How much more likely do you feel that the robot may spill the coffee while using an umbrella?*
- **Q2:** *How much do you care about the robot being wet?*

The first question is designed to elicit the participant's belief about the robot's domain dynamics while the second question is for the participant's preference. Their feedback

Question	p-value	Mobile	Humanoid
(Q1) Domain Dynamics	0.047	2.92	3.58
(Q2) Weight Preference	0.027	2.83	3.67

Table 2. Averaged participants' responses to the two questions for each setting before viewing the demonstrations. They are asked in a 5-point Likert scale where 1 is the lowest and 5 the highest.

for each setting is presented in Table 2. The participants of the mobile robot setting believed that the robot would be less likely to spill the coffee while holding the umbrella than the participants of the humanoid setting. *Notice that this is in contrast to the ground truth.* Meanwhile, the participants expressed more concern about the robot getting wet in the humanoid setting than the mobile robot setting.

After the questions, we asked the participants to rate the demonstrations. Accordingly, we find that the ratings for the demonstrations where the robot operates in the rain without an umbrella, or takes an umbrella in a sunny day to be rated low in the humanoid setting. In contrast, in the mobile setting, fewer demonstrations received low ratings. These results supported our assumption that the human are easily biased when working with robots.

Next, we run our method under each setting to see whether our method can recover from such biased beliefs. For comparison, we also run GeReL^{-}, which performed similarly to MaxEnt-IRL in our simulation task. We run each method for each participant in both settings. The ratings are normalized to remove inconsistencies across different participants. The results are presented in Figure 12. We observed that the learned probability of spilling coffee while holding an umbrella by GeReL for the humanoid robot setting is generally larger than the mobile robot setting. This represents the estimated human understanding of the domain dynamics, which is consistent with the participant's feedback shown in Table 2. Furthermore, GeReL

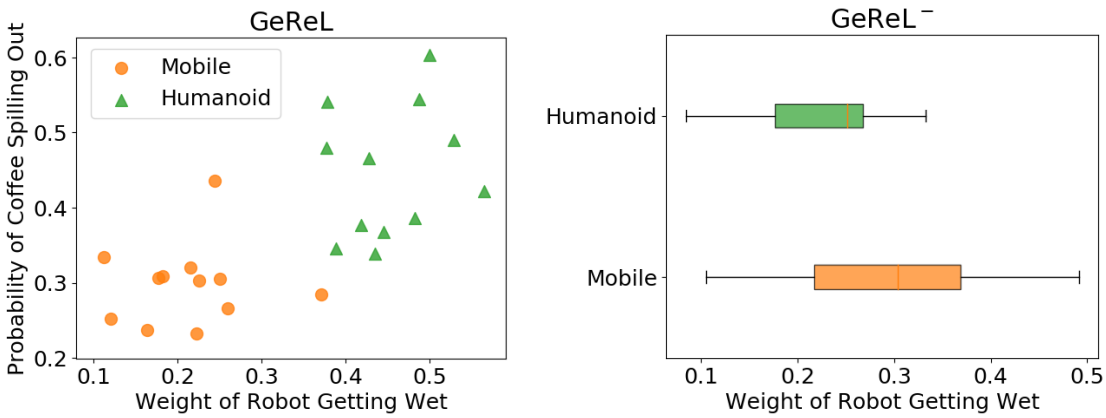


Figure 12. Feature weights learned by GeReL and GeReL^{-}.

learned that the participants cared more about the robot getting wet in the humanoid setting than the mobile robot setting, which is also consistent with the participant's true preference. In contrast, GeReL^{-} discovered just the opposite!

2.4 Conclusion

In this work, we looked the Generalized Reward Learning (GRL) problem and proposed a method called GeReL to address it. GeReL removes the assumption that the human always maintains the true belief about the agent's domain dynamics. To develop the method, we formulated the GRL problem in a variational inference framework to infer the parameters governing the reward function and the human's belief about the agent simultaneously. To reduce the effort for obtaining training samples, we used the human's ratings of agent demonstrations. We evaluated our approach experimentally using a simulated domain and with a user study. The results showed that GeReL outperformed prior approaches that could have misinterpreted the human preferences when such biases are not considered. We showed that GeReL

could recover the true human preferences effectively even under such a challenging setting.

Once the biased domain dynamics is obtained, the next question is how to use it. The simplest method of course is to inform the human about his biases and hope that it would work. An alternative method that is often considered in the area of human-aware planning is that the agent could, instead of always pursue optimal behaviors, behave to match with the human's expectation whenever feasible, so as to behave in an explicable manner. In contrast to the multi-objective MDP problem (Roijers et al. 2013; Chatterjee, Majumdar, and Henzinger 2006) which has more than one reward function to optimize, in this problem, the agent maintains two transition functions, one for its own dynamics and the other for the human's belief of it. There already exists work that looks at this problem (Zhang et al. 2017; Chakraborti, Sreedharan, et al. 2017).

Like all reward learning problems, the solution is not unique. This is commonly known as the non-identifiability issue. In general reward learning (GRL), an additional complexity is the learning of the transition function, which unfortunately only aggravates the issue. So far, we are not aware of any solutions to this problem except for the ones that introduce inductive biases on the priors or the error functions, such as Bayesian IRL (Ramachandran and Amir 2007) and apprenticeship learning methods (Abbeel and Ng 2004). In this regard, our work also introduces an inductive bias by assuming a form of the posterior as a multivariate Gaussian distribution.

In terms of simultaneously learning different factors, there exist prior results (Armstrong and Mindermann 2018) that argue against it and prove that it is impossible to determine one without assuming some form of the other. However, we note that the negative results apply only to the situation where one of the factors is the compu-

tational process. Consider the function $C(R, M) = \Gamma$. When C is given, the choices of $r \in R$ and $m \in M$ are connected to the corresponding value of $\gamma \in \Gamma$. However, if only m is given, we may choose any r and then simply remap (choosing a $c \in C$) (r, m)'s to their corresponding γ's. This flexibility of the computational process is the core reason of the negative results. However, the non-identifiability issue is still there.

Chapter 3

NEUTRALIZING THE IMPACT OF BIASED BELIEF IN PREFERENCE-BASED REINFORCEMENT LEARNING

3.1 Introduction

Reinforcement learning agents learn by interacting with the environment and optimizing its behavior in terms of the received reward signals. Providing proper reward signals is critical to the agent learning the right behavior. However, manually designing a reward function that expresses what the human really want is challenging, especially for complex domains and tasks. Instead of relying on expensive and vulnerable reward engineering, Preference-based Reinforcement Learning (PbRL) (Akrour, Schoenauer, and Sebag 2011; Wilson, Fern, and Tadepalli 2012; Busa-Fekete et al. 2013, 2014; Wirth and Fürnkranz 2013c; Christiano et al. 2017; Lee, Smith, and Abbeel 2021) replaces the requirement of hand-engineered reward functions with human preferences between the agent's behaviors, which are easier to solicit. Typically, a surrogate reward function is learned to be consistent with the observed preference, then leveraged to optimize the intelligent agent's policy. The existing PbRL literature generally builds on the insight that the human provides feedback depending on her expectation (i.e., the policy the human thinks the agent should follow). Thus, the introduced surrogate reward function actually describes the human expectation, rather than the human reward function. The human expectation can be influenced by both her reward function and belief about the domain dynamics. The approaches may work in cases where the human belief doesn't affect the human's expectation and

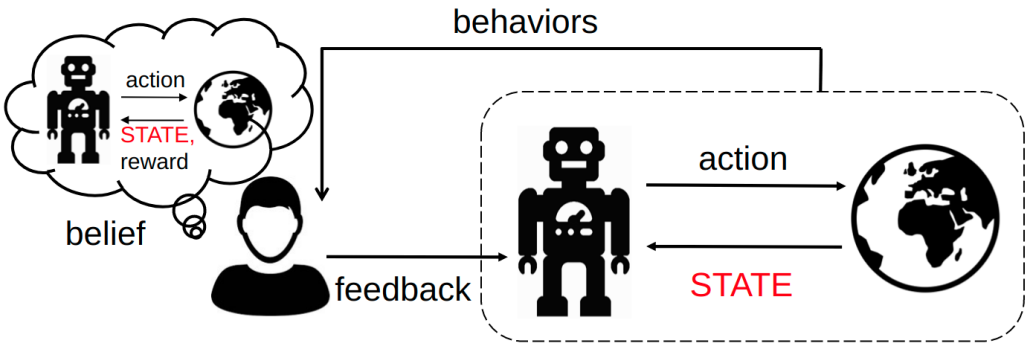


Figure 13. The human user observes the robot's behavior and provides preferences based on her own reward model and belief about the agent's domain dynamics, which may be different from the true dynamics. Our goal is to infer the reward function while neutralize the impact of such bias, thereby obtain an optimal policy.

judgement much. In many scenarios, however, the human holds a biased belief about the domain dynamics which could bring about biased feedback about the agent's behaviors. Thereby, it would mislead the reward learning, result in suboptimal and degraded performance, even introduce unintended consequences. Thus it is improper to directly use the learned surrogate reward function as the reward function, especially in cases the human belief about the domain dynamics are biased.

Consider a mobile robot that is tasked to deliver packages on campus. When it has to cross the street, the optimal behavior is to go through the pedestrian crossing. But, a human user may have a biased belief about the agent's capability (i.e., dynamics) such that she thinks the agent are poorly capable of avoiding collision when there are lots of people around. Thus, the human would prefer the agent going through the pedestrian bridge where there are fewer people, but further away and would cost more energy. According to the human preference, the agent may misunderstand what the human really wants and infer that it will gain more rewards by going through the pedestrian bridge. In such cases, the agent would learn a policy that chooses to use the pedestrian bridge all the time, even it is adept in avoid obstacles and this policy

is quite inefficient and will cost much more energy. To address issues like this, the learning agents must have the capability to consider the human belief while learning the reward function and neutralize the impact of the human's biased belief about domain dynamics in PbRL.

In this work, we assume the human is noisily rational at generating her expectations. The human expectation are generally determined by the human rewards and belief about the domain dynamics (Reddy, Dragan, and Levine 2018; Gong and Zhang 2020). Therefore, for the scenarios where the human's belief is biased (i.e., deviation from the true domain dynamics) which is very likely to happen in the real world (Kahneman 2011; Herman et al. 2016; Reddy, Dragan, and Levine 2018; Gong and Zhang 2020), the biased belief would lead to biased feedback that can misguide the learning process, resulting in slow convergence, sub-optimal behaviors, and potentially safety risks to humans. The problem setting is illustrated in Figure 13. In this work, we formulate the problem as PbRL under human biased belief about domain dynamics (PrefBias). Similar to prior work (Wirth and Fürnkranz 2013c; Christiano et al. 2017; Lee, Smith, and Abbeel 2021), the agent aims to infer a surrogate reward function from the human's preference between pairs of segments of the agent's behavior. What distinguishes our work from the prior work is that our goal is to learn a reward function with consideration of the human's biased belief. The key challenge is that we need to learn two models (i.e., reward and belief) instead of reward function only. We model both the reward function and human's belief using Bayesian neural networks (BNN) in order to capture the uncertainty and take advantage of potential informative priors. Given the human's preferences, we assume the preferences are generated in terms of both models following the Bradley-Terry framework (Bradley and Terry 1952) and update the model parameters alternately via a variational Bayesian

inference framework (Blundell et al. 2015; Kingma, Salimans, and Welling 2015). We assume that the human's belief about the dynamics can be treated as disturbed true domain dynamics. An estimated dynamics model is learned by interacting with the environment (Kurutach et al. 2018; Clavera et al. 2018; Wang et al. 2019) and serves as the prior of the human's belief. To get informative queries, we select queries that are predicted most differently with respect to agent's and human's model. The agent's policy is optimized within a standard reinforcement learning framework with respect to the learned reward function. Note that, the estimated human biased belief is not used for policy learning, and it is utilized to assist in inferring the surrogate reward function. The agent's policy is determined by the learned reward function and true domain dynamics.

The main contribution of this work lies in generalizing PbRL under biased human belief, which is common in real-world tasks. We first validate our method on two 2D navigation domain in which the human's preference is synthesized according to manually introduced models with biases. It is then tested on a modified lunar lander domain (Brockman et al. 2016) and two typical locomotion domain from PyBullet (Coumans and Bai 2021), with synthesized human models as well. We compare PrefBias with a state-of-the-art PbRL method, unsupervised PrEtraining and preference-Based learning via relaBeLing Experience (PEBBLE) (Lee, Smith, and Abbeel 2021) with synthetic human feedback. A state-of-the-art RL method, Soft Actor Critic (SAC) (Haarnoja, Zhou, Abbeel, et al. 2018; Haarnoja, Zhou, Hartikainen, et al. 2018) with the true reward function is applied to provide us oracle performance. The results show that PrefBias outperforms PEBBLE and is comparable to SAC in terms of trajectory returns. Our method successfully recovers the agent's optimal

policies while the human feedback is provided under biased belief. We also investigate the influences of different priors and query sampling approaches.

3.2 Related Work

Preference-based Reinforcement Learning (PbRL) has been successfully applied in various tasks (Wirth and Fürnkranz 2013c). The preferences can be used to learn a policy directly via estimating a distribution of parametric policy space (Wilson, Fern, and Tadepalli 2012), or to compare and rank policies (Busa-Fekete et al. 2013, 2014). Moreover, a preference model can be learned, then it is used to get a ranking for actions given a state, then derive a policy (Hüllermeier et al. 2008; Wirth and Fürnkranz 2013a, 2013b). However, these approaches suffer from feedback efficiency issues such that it needs many human preference data to achieve near-optimal policy. Another line of PbRL is to learn a surrogate reward function which can be used to optimize the policy. Christiano et al. (Christiano et al. 2017) models the reward function using deep neural networks which scale it to much complex tasks. PEBBLE (Lee, Smith, and Abbeel 2021) is proposed recently to improve the sample and feedback efficiency by unsupervised pretraining and off policy learning with reward relabeling. The surrogate reward function, however, describes the human expectation, rather than the reward function. We interpret the human expectation as determined by the human reward function and her belief about the domain dynamics. While the belief deviates from the true domain dynamics and impact the human feedback being biased, especially under non-expert settings, it may lead the agent to learning a wrong reward function and result in degraded performance, even undesired behavior.

Researchers have been investigating the model differences between human and

agent in the form of domain dynamics (Zhang et al. 2017; Chakraborti, Kambhampati, et al. 2017; Chakraborti et al. 2019; Reddy, Dragan, and Levine 2018; Gong and Zhang 2020). Formulated in the form of classical planning setting, Chakraborti et al. (Chakraborti, Sreedharan, et al. 2017; Kulkarni et al. 2019) leverage the differences between the domain models to generate explanations to the human. But, in these work, the human belief model is assumed to be given a prior which is impractical for the real world problems. Reddy et al. (Reddy, Dragan, and Levine 2018) manages to learn the human belief model in the presence of reward function using inverse soft Q-learning method. Then the learned belief model is used to assist human decision making. Furthermore, GeReL (Gong and Zhang 2020) uses a variational inference method to infer both the reward function and human belief in simple discrete domains by soliciting the human's ratings of the agent's behavior, which would be used to generate the agent's policy. Our work has a similar problem setting as in (Gong and Zhang 2020), but uses human preferences between agent's behavior, and focuses on continuous domain. In our work, the reward and human belief are modeled using BNNs such that it can scale to complex continuous control tasks. We aim to infer the reward function from human feedback while consider the human belief about the domain dynamics meanwhile.

3.3 Our Approach

3.3.1 Preliminaries

Reinforcement learning requires access to a reward function that incentive the agent to learn right behavior. However, the reward function is difficult to specify,

especially for complex domain and task. Without the hand-engineered reward function, PbRL provides an alternative: the agent solicits a human teacher's preference between a pair of behaviors and learn a surrogate reward function that is consistent with the observed feedback. The learned surrogate reward function is then be used to optimize the agent's policy.

3.3.1.0.1 Deep Reinforcement Learning from Human Preferences

Christiano et al. (Christiano et al. 2017) proposed a framework that models the probability that the human prefers one behavior segment over another as proportional to exponentiated sum of a surrogate reward function $\hat{r}$, following the Bradley-Terry framework (Bradley and Terry 1952) as in Equation (3.1).

$$\hat{p}(\tau_1 \succ \tau_2) = \frac{\exp \sum \hat{r}(s_t^1, a_t^1)}{\exp \sum \hat{r}(s_t^1, a_t^1) + \exp \sum \hat{r}(s_t^2, a_t^2)}, \tag{3.1}$$

where τ is a sequence a states and actions $\{(s_1, a_1), \ldots (s_t, s_t)\}$. $\tau_1 \succ \tau_2$ indicates the event that segment τ_1 is preferable to segment τ_2. The surrogate reward function $\hat{r}$ is learned by minimizing the cross entropy loss using the observed preference data. The agent then optimize its policy to maximize the discounted sum of $\hat{r}$ using existing RL algorithms. Consider the surrogate reward function $\hat{r}$ may be non-stationary since it is updated during learning, Christiano et al. (Christiano et al. 2017) used on-policy RL algorithm PPO for policy optimization.

3.3.1.0.2 PEBBLE

In order to improve the sample and feedback efficiency, Lee et al. (Lee, Smith, and Abbeel 2021) introduce a new PbRL framework PEBBLE that improves Christiano et al. (Christiano et al. 2017) work from two main aspects:

- Unsupervised exploration: Before collecting human feedback, the agent is pre-trained using intrinsic motivation (Oudeyer, Kaplan, and Hafner 2007; Schmidhuber 2010) (i.e., the state entropy $\mathcal{H}(s) = -\mathbb{E}_{s\sim p(s)}[\log p(s)]$) to learn how to generate diverse behaviors. Thus, the agent's behavior could have a better state coverage and collect more informative human feedback.
- Off-policy RL with reward relabeling: PEBBLE proposed to use a state-of-the-art off-policy RL algorithm SAC (Haarnoja, Zhou, Abbeel, et al. 2018; Haarnoja, Zhou, Hartikainen, et al. 2018) to improve the sample-efficiency of on-policy RL algorithm. Moreover, to overcome the non-stationary problem in reward learning, PEBBLE relabels all the past experiences every time it gets the updated surrogate reward function.

3.3.2 Problem Formulation

Existing PbRL methods endow the intelligent agent with the ability of learning from human preferences that convey the information of the policy the human teacher wants the agent to follow. Therefore, the observed preferences describes the human's expectation, rather than the reward function. We interpret the expectation as determined by both human's reward function and her belief about the domain dynamics. When the belief is biased (i.e., deviates from the true domain dynamics), the human

teacher may provide preferences that mislead the agent to learn a wrong reward function. Thereby, optimizing such a reward function may result in degraded performance and unintended behaviors. In this work, we remove the restrictive assumption that the human always maintains a correct belief about domain dynamics and aim to learn a reward function that captures what the human really wants while taking biased human belief into consideration simultaneously. We formalize the problem as follows:

Given:

- Agent's rollouts in the environment,
- Human's preference data for pairs of behavior segments of the agent.

To determine:

- A surrogate reward function,
- Estimation of human's belief about the domain dynamics,
- Agent's policy that optimizes the surrogate reward function under the true domain dynamics.

To solve the problem, we propose a method called *PbRL under biased belief about domain dynamics (PrefBias)*. The workflow of our approach is demonstrated in Figure 14. The human's preference between a pair of behavior segments would be determined not only by the cumulative rewards but also the occurrence probability of the agent's behavior with respect to the human belief. The surrogate reward function and belief model are estimated alternately from the human's preference data. Finally, the agent's policy is optimized in terms of the learned reward function.

3.3.3 Methodology

The environment is formulated as Markov Decision Processes (MDPs). MDPs is a tuple, $(\mathcal{S}, \mathcal{A}, f, r, \gamma, \rho)$, where $\mathcal{S}$ is a set of states of the environment, and $\mathcal{A}$ denotes a

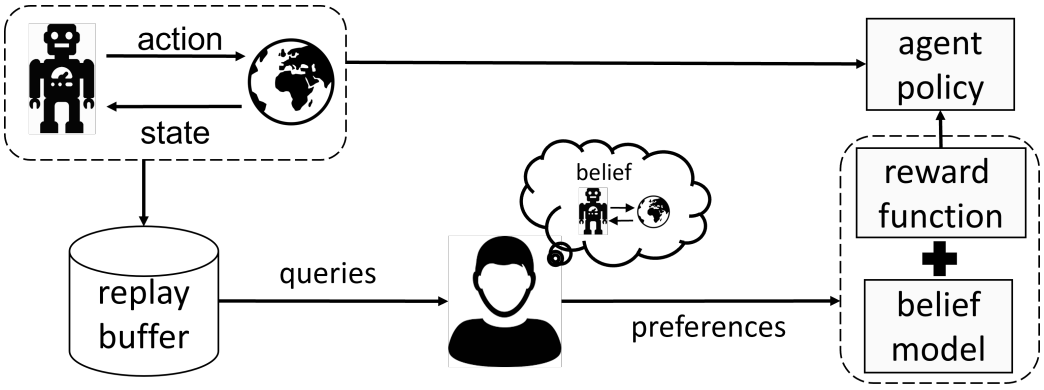


Figure 14. Workflow of our approach. Intelligent agent collects data by interacting with environment. We then select a set of trajectory segments to solicit the human preferences about the agent's behaviors which will be used to learn a surrogate reward function and human belief about domain dynamics. The agent's policy is optimized using the learned reward function.

set of actions. $f : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{S}$ describes the domain dynamics, $r : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is a reward function. Discount factor γ determines how the agent favors current rewards over future rewards. ρ is a distribution over the initial state. In the following sections, we will talk about three main components of our method: inference from human preference; human preference modeling; and queries selection.

3.3.3.1 Inferring Reward function and Human Belief Simultaneously

We aim to learn a surrogate reward function while inferring the human belief at the same time. We use $\hat{r}_w(s, a)$ and $\hat{f}^H_\phi(s, a)$, which are parameterized by w and ϕ, to characterize the reward function and the human's belief about the domain dynamics, respectively. Given the observed human preference D_{pref}, our goal is to infer the posterior distribution of the model parameters $p(w, \phi|D_{pref})$. However, it is intractable to learn the posterior directly since it requires the computation of the integral of $p(D_{pref})$. We introduce a variational posterior $q(w, \phi; \Omega)$ to approximate

the true posterior distribution, where Ω is a set of parameters governing w and ϕ. In this work, we assume both w and ϕ follow Gaussian distributions such that Ω consists of means and standard deviations of w and ϕ, i.e., $\Omega = (\mu_w, \sigma_w, \mu_\phi, \sigma_\phi)$. To capture the uncertainty in the model parameters, $\hat{r}_w(s,a)$ and $\hat{f}^H_\phi(s,a)$ are modeled using two Bayesian Neural Networks (BNN). For $\hat{r}_w(s,a)$, the output is a real value. For $\hat{f}^H_\phi(s,a)$, the network is two-headed as we assume each transition follows a Gaussian distribution. It outputs the mean and standard deviation of the next state given s and a instead of an exact estimation. Similar as in the work of Bayes By Backprop (Blundell et al. 2015), we aim to learn a variational posterior distribution that is close to the true posterior which is measured by the Kullback–Leibler (KL) divergence:

$$\Omega^* = \arg\min_{\Omega} \mathcal{D}_{\mathrm{KL}} \left[q(w, \phi; \Omega) \| p(w, \phi | D_{pref}) \right].$$

The objective function can be written as:

$$\begin{aligned} \mathcal{F}(\Omega, D_{pref}) &= \mathcal{D}_{\mathrm{KL}} \left[q(w, \phi; \Omega) \| p(w, \phi | D_{pref}) \right] \\ &= \mathcal{D}_{\mathrm{KL}} \left[q(w, \phi; \Omega) \| p(w, \phi) \right] - \mathbb{E}_{q(w,\phi;\Omega)} \left[\log p(D_{pref} | w, \phi) \right]. \end{aligned} \tag{3.2}$$

We assume w and ϕ are independent from each other such that $q(w, \phi; \Omega) = q(w; \Omega_w) q(\phi; \Omega_\phi)$ where $\Omega_w = (\mu_w, \sigma_w)$ and $\Omega_\phi = (\mu_\phi, \sigma_\phi)$. The Equation (3.2) then becomes:

$$\begin{aligned} \mathcal{F}(\Omega, D_{pref}) &= \mathbb{E}_{q(w;\Omega_w)q(\phi;\Omega_\phi)} \left[\log q(w; \Omega_w) q(\phi; \Omega_\phi) - \log p(w) p(\phi) \right] \\ &\qquad - \mathbb{E}_{q(w;\Omega_w)q(\phi;\Omega_\phi)} \left[\log p(D_{pref} | w, \phi) \right] \\ &= \underbrace{\mathbb{E}_{q(w)} \left[\log q(w) - \log p(w) \right]}_{\mathcal{D}_{\mathrm{KL}}[q(w) \| p(w)]} + \underbrace{\mathbb{E}_{q(\phi)} \left[\log q(\phi) - \log p(\phi) \right]}_{\mathcal{D}_{\mathrm{KL}}[q(\phi) \| p(\phi)]} - \underbrace{\mathbb{E}_{q(w)q(\phi)} \left[\log p(D_{pref} | w, \phi) \right]}_{log-likelihood}. \end{aligned} \tag{3.3}$$

For brevity, we omit the parameters of the variational posteriors. In Equation (3.3), the first two terms are KL divergence between the variational posterior distributions and

the priors for w and ϕ respectively. The last term is the expectation of log-likelihood of D_{pref} in terms of variational posteriors. In order to update the parameters of our posterior distributions, we take the gradients for Ω_w and Ω_ϕ separately:

$$\begin{aligned}
&\nabla_{\Omega_w}\mathcal{F}(\Omega, D_{pref}) \\
&\quad = \nabla_{\Omega_w}\mathbb{E}_{q(w;\Omega_w)}\left[\log q(w;\Omega_w) - \log p(w)\right] - \nabla_{\Omega_w}\mathbb{E}_{q(w;\Omega_w)q(\phi;\Omega_\phi)}\left[\log p(D_{pref}|w,\phi)\right],
\end{aligned}$$

and similarly,

$$\begin{aligned}
&\nabla_{\Omega_\phi}\mathcal{F}(\Omega, D_{pref}) \\
&\quad = \nabla_{\Omega_\phi}\mathbb{E}_{q(\phi;\Omega_\phi)}\left[\log q(\phi;\Omega_\phi) - \log p(\phi)\right] - \nabla_{\Omega_\phi}\mathbb{E}_{q(w;\Omega_w)q(\phi;\Omega_\phi)}\left[\log p(D_{pref}|w,\phi)\right],
\end{aligned}$$

where $p(w)$ and $p(\phi)$ are the priors in the form of Gaussian distributions as well. To overcome the problem of high variance in learning process, the trick of local reparameterization (Kingma, Salimans, and Welling 2015) is applied.

Prior. We have no information about the reward function. Thus, the mean of $p(w)$ is initialized randomly using a uniform distribution. When informative knowledge of the reward function is given, the prior can also be easily set to convey the expert knowledge. For the belief about the domain dynamics, other than random initialization, we explore an alternative prior. We assume that the human's belief about the agent's dynamics can be treated as disturbances of the true dynamics. Hence, we set the mean of $p(\phi)$ to be the parameters of an estimated true dynamics model during the inference. Similarly, any other priors of human's belief can also be utilized into the learning paradigm by leveraging distinct domain knowledge.

3.3.3.2 Human Preference Model

The log-likelihood of the human preference data can be rewritten as the summation of the log-likelihood of each instance d in the data set, i.e., $\log p(D_{pref}|w,\phi) = \sum \log p(d|w,\phi)$. For $p(d|w,\phi)$, we attribute the human preference between two segments (i.e., τ_1 and τ_2) to the difference between the evaluation of each segment. Following the Bradley-Terry model (Bradley and Terry 1952), the human preference model is formulated using $\hat{f}_\phi^H$ and $\hat{r}_w$ as:

$$\hat{p}(\tau_1 \succ \tau_2|w,\phi) = \frac{\hat{p}(\tau_1|w,\phi)}{\hat{p}(\tau_1|w,\phi) + \hat{p}(\tau_2|w,\phi)}, \tag{3.4}$$

where $\hat{p}(\tau)$ accounts for the probability of occurrence of the trajectory τ and is defined in the form of probabilistic view of RL (Levine 2018) that consists of both $\hat{f}_\phi^H$ and $\hat{r}_w$:

$$\hat{p}(\tau) \propto \rho(s_0) \left(\prod_t \hat{f}_\phi^H(s_{t+1}|s_t,a_t) \right) \exp\left(\sum_t \hat{r}_w(s_t,a_t) \right). \tag{3.5}$$

The main differences between Equation (3.5) and the preference model proposed in Equation (3.1) is that, rather than $\exp\left(\sum_t \hat{r}_w(s_t,a_t)\right)$ solely, we assume that the human's assessment of the agent's behavior will also affected by her belief about the agent's dynamics $\hat{f}_\phi^H$. For reward learning, we fit the human data to the preference model and optimize the cross entropy loss as in (Christiano et al. 2017):

$$\text{loss}(\hat{r}) = -\sum_{(\tau_1,\tau_2,\mu)\in D} \mu(1) \log \hat{p}[\tau_1 \succ \tau_2] + \mu(2) \log \hat{p}[\tau_2 \succ \tau_1].$$

3.3.3.3 Queries Selection

During training, the agent collects transitions D_{dyna} by interacting with the environment. The queries for soliciting human's preferences is then selected from D_{dyna}.

Algorithm 2 PbRL under biased belief about domain dynamics (PrefBias)

Initialize the robot's policy π_θ, robot dynamics $\hat{f}_\psi(s,a)$, human's belief about dynamics $\hat{f}^H_\phi(s,a)$, reward function $\hat{r}_w(s,a)$ and empty sets D_{dyna} and D_{pref}.
repeat
  Collect samples from environment f using π_θ and add them to D_{dyna}.
  Train $\hat{f}_\psi(s,a)$ using D_{dyna}.
  repeat
    Actively select K pairs of segments from D_{dyna} to solicit human preferences and add them to D_{pref}.
    Infer $\hat{f}^H_\phi(s,a)$ and $\hat{r}_w(s,a)$ using human's preference data D_{pref} with $\hat{f}_\psi(s,a)$ to be the prior of $\hat{f}^H_\phi(s,a)$.
  until present a predefined number of samples
  Update π_θ using SAC with $\hat{r}_w(s,a)$.
until the policy performs well in real environment f

To mitigate the burden on the human, we should select queries that maximize the information we can receive from the human's feedback. In prior PbRL work (Christiano et al. 2017; Lee, Smith, and Abbeel 2021), several query sampling approaches has been explored, such as uniform sampling, disagreement-based sampling, and entropy-based sampling:

- Uniform sampling: We pick K pairs of segments uniformly at random from the transition buffer.
- Disagreement-based sampling: We first generate the initial batch of $3 \times K$ pairs of segments uniformly at random, measure the variance across ensemble of preference predictors (Equation (3.4)), and then select the top K from them.
- Entropy-based sampling: Similarly as disagreement-based sampling, we first generate the initial batch of $3 \times K$ pairs of segments uniformly at random, measure the entropy of a single preference predictor (Equation (3.4)), and then select the top K pairs of segments with high entropy.

In this work, we explore an additional method (i.e., preference disagreement-based sampling) that relies on both agent's model and human belief. To find queries that better elicit the information of the human biased belief model, we consider the queries that have most different preferences estimated using both the agent's model and learned human belief model during the training process,

$$q_n^* = \arg\max_{q_n} \mathcal{D}_{\mathrm{KL}} \left[\hat{p}(q_n; \hat{r}_w, \hat{f}_\psi) \| \hat{p}(q_n; \hat{r}_w, \hat{f}_\phi^H) \right]$$

where $\hat{r}_w$ is the surrogate reward function, $\hat{f}_\psi$ in the estimated true dynamics model, and $\hat{f}_\phi^H$ is the learned human belief. $\hat{p}$ is the probability that the human would prefer one behavior segment over another. It is computed using Equation (3.4) with the given models.

3.3.3.4 Algorithm Overview

The algorithm is presented in Algorithm 2. We define the agent's policy as $\pi_\theta : \mathcal{S} \mapsto \mathcal{A}$ which is parameterized by θ. The data is collected by interacting with the environment using π_θ. In order to investigate the performance of using true domain dynamics as the prior of human belief model, we need to learn an estimated model from the environment. The dynamics approximator is presented as a neural network $\hat{f}_\psi(s_t, a_t)$ which is parameterized by ψ. The dynamics model is learned by minimizing L_2 one-step prediction loss.

$$\min_\psi \frac{1}{|D_{dyna}|} \sum_{(s_t, a_t, s_{t+1}) \in D_{dyna}} \left\| s_{t+1} - \hat{f}_\psi(s_t, a_t) \right\|_2^2,$$

where $D_{dyna} = \{(s_t, a_t, s_{t+1}), \dots\}$ is the training data set that stores the agent transitions. We use Adam optimizer (Kingma and Ba 2014) to solve this supervised learning problem. To avoid overfitting, several standard techniques are applied, such

as using validation set for early stopping, and normalization for inputs and outputs of the network (Clavera et al. 2018; Kurutach et al. 2018). To make it feasible that $\hat{f}_\psi(s,a)$ being the prior of $\hat{f}^H_\phi(s,a)$, it has the same architecture of $\hat{f}^H_\phi(s,a)$. D_{pref} is used to learn the surrogate reward function $\hat{r}_w(s,a)$ and human's belief about the domain dynamics $\hat{f}^H_\phi(s,a)$. The agent then optimizes its policy in terms of $\hat{r}_w(s,a)$ and $\hat{f}_\psi(s,a)$. For policy learning, a state-of-the-art RL method, Soft Actor Critic (SAC) (Haarnoja, Zhou, Abbeel, et al. 2018; Haarnoja, Zhou, Hartikainen, et al. 2018) is applied to learn the agent's policy π_θ. We tried both Proximal Policy Optimization (PPO) (Schulman et al. 2017) and SAC. The latter gives us better performance. Note that, to ensure σ_w and σ_ϕ are always non-negative, we parameterize them with ρ_w and ρ_ϕ and transform σ_w and σ_ϕ with the softplus function (Blundell et al. 2015), $\sigma = \log(1 + \exp(\rho))$.

3.4 Evaluation

We evaluate the performance of our method on a set of continuous controls tasks. The goal of the experiments is to verify that our method can successfully infer a surrogate reward function that results in a policy with higher trajectory returns and being close to the oracle performance with the ground truth reward function. Moreover, the baseline PbRL method is hardly recovering a policy that performs efficiently in the environment as measured by the ground truth reward function.

3.4.1 Experiments Setups

3.4.1.1 Learn from Simulated Human Models

Similar as the standard problem setting of PbRL (Wirth and Fürnkranz 2013c; Akrour, Schoenauer, and Sebag 2011; Busa-Fekete et al. 2013, 2014; Christiano et al. 2017; Lee, Smith, and Abbeel 2021), the agent interacts with the environment to collect the trajectories, but have no access to the underlying reward signals. It has to solicit the human's preference between segments of the agent's trajectories, and use the human feedback to infer a surrogate reward function. In our work, we assume the human evaluates the agent's behavior based on both her reward function and belief about the agent's dynamics, rather than rewards only. Hence, we design a human belief model which is different from the true domain dynamics. A synthetic human teacher provides preference based on the ground truth reward function and the fictitious human belief model. Using the simulated human model, we can quantitatively verify the efficacy of our method compared to alternative approaches.

3.4.1.2 Baselines

For evaluation, we compare our method with two baselines.

- **Soft Actor Critic (SAC)** (Haarnoja, Zhou, Abbeel, et al. 2018; Haarnoja, Zhou, Hartikainen, et al. 2018). It is a state-of-the-art off-policy RL algorithm. We run SAC with the underlying ground truth reward signals from the environment. It would provide us the oracle performance.
- **unsupervised PrEtraining and preference-Based learning via rela-**

BeLing Experience (PEBBLE) (Lee, Smith, and Abbeel 2021). This is a state-of-the-art PbRL approach. It builds upon the work of (Christiano et al. 2017). In order to improve the sample and human feedback efficiency, the authors pre-train the policy to be learned with intrinsic motivation to explore, and relabel the replay buffer with previous learned reward to mitigate the effects of a non-stationary reward learning. It implicitly assume the human has a correct belief about the domain dynamics. Hence, it may learn a wrong reward function from the human preference which are affected by the biased belief.

3.4.2 Benchmark Tasks

3.4.2.1 2D Navigation Task

We start with two 2D navigation tasks as shown in Figure 15a and 15b. The gray areas are walls. For domain Navigation 1, we consider three types of terrains, such as dirt, ice, and sand. There are two corridors in the environment separated by a wall in the center. The agent is at the center right initially. Its task is to navigate to the goal area (green) at the center left. The upper corridor (blue) has ice on it and the lower corridor (yellow) is sandy. The state space has two variables: the coordinates of the agent in x and y. The action space specifies the velocity and orientation of each action of the agent. The agent would receive a reward of $+10$ as it reach the goal. There is also a living reward of -1 for each time step. In this domain, the human's belief about the capability of the agent handling different types of terrains may be different from the true dynamics. Thereby biased feedback would be provided subject

to the biased belief, and mislead the reward learning. We design the true dynamics and simulated human belief as follows.

- **True Dynamics:** The agent is capable of handling the icy terrain, but may be stuck in the sand.
- **Human Belief:** The agent's behavior would be much more stochastic due to the slippery road condition, and dealing with the sandy terrain well although being slow sometime.

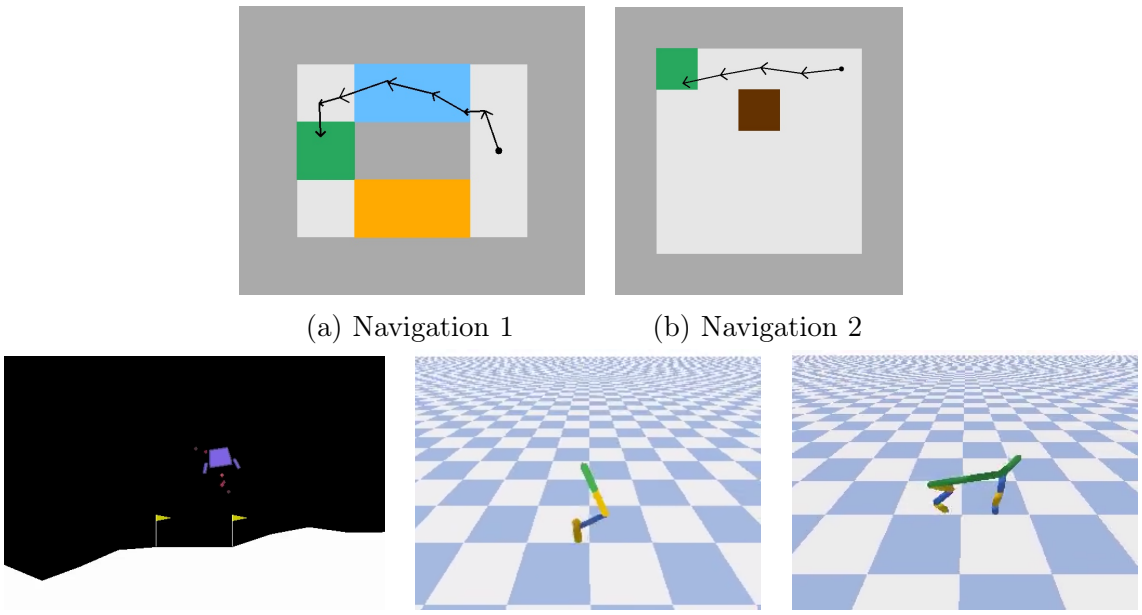
For Navigation 2, there is a pit in the center which is indicated in brown. The agent would be penalized by -10 as reaching the pit. The agent starts from the upper right of the map. Its task is to navigate to the goal area in upper left. In this domain, the human has a biased belief about dynamics where the agent is approaching the pit area.

- **True Dynamics:** The agent is capable of moving nearby the pit area.
- **Human Belief:** The agent may fall into the pit when it is getting closer to it.

With this biased belief, the human would have a different expectation of the agent's behavior and prefer the suboptimal trajectories.

3.4.2.2 Lunar Lander Task

This is classical control task from OpenAI gym (Brockman et al. 2016). The lander is tasked to land as close to the target landing pad between two flag poles as possible, making sure that both side boosters are touching the ground. If the lander moves away from landing pad it loses reward. Episode finishes if the lander crashes



(a) Navigation 1 (b) Navigation 2

(c) Lunar lander (d) Hopper (e) Cheetah

Figure 15. Illustration of the domains we test on.

or comes to rest. The observation space consists of 8 states: the coordinates of the lander in $x\&y$, its linear velocities in $x\&y$, its angle, its angular velocity, and two booleans that represent whether each leg is in contact with the ground or not. In this work, we conduct experiments on the continuous version of lunar lander. The action space is described by two factors. The first coordinate of an action determines the throttle of the main engine, while the second coordinate specifies the throttle of the lateral boosters. Reward for moving from the top of the screen to the landing pad and coming to rest is about 100-140 points. If the lander moves away from the landing pad, it loses reward. If the lander crashes, it receives an additional -100 points. If it comes to rest, it receives an additional +100 points. Each leg with ground contact is +10 points. Firing the main engine is -0.3 points each frame. Firing the side engine is -0.03 points each frame. Solved is 200 points (*Gym Documentation*). In order to test

our method, we modify the domain dynamics of which the human has no information and would raise biases.

- **True Dynamics:** There is invisible solar wind blowing from left to right. Wind is simulated by changing a percentage of the left action to no-op and reinforcing the right action.
- **Human Belief:** In the space, the lander operate in a vacuum. The lander traces are only controlled and influenced by its engines.

The human is not aware of the solar wind in the environment, thus having a biased belief about the domain dynamics in that area. It could result in learning a wrong reward function using the human preference between behavior segments if we don't take the biased belief into consideration.

3.4.2.3 Locomotion Tasks

We consider two typical locomotion tasks developed using the Bullet physical simulator (Coumans and Bai 2021): Hopper and Cheetah (see Figure 15d and 15e). The planar one-legged hopper introduced in (Lillicrap et al. 2015) has a state space of 14 dimensions and an action space of 4 dimensions. It is tasked to move forward and rewarded for torso height and forward velocity. The cheetah is introduced based on the work by (Wawrzyński 2009; Wawrzyński and Tanwani 2013). It has a state space of 18 dimensions and an action space of 6 dimensions. The agent is tasked to move forward as quickly as possible with a cheetahlike body that is constrained to the plane. The reward is linearly proportional to the forward velocity up to a maximum of 10m/s.

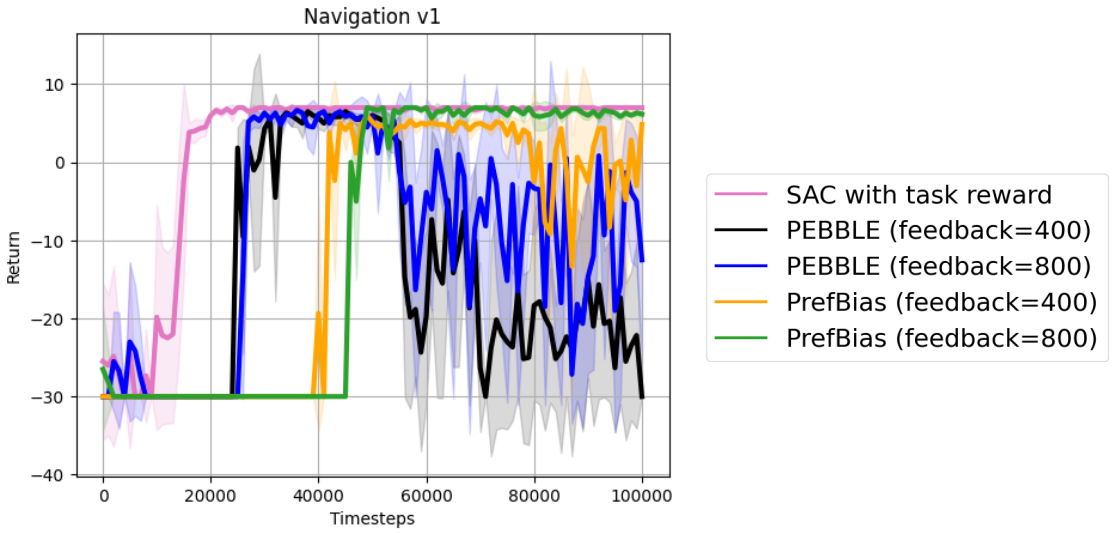


Figure 16. Learning curves on the task Navigation 1 as measured on the ground truth reward.

In this work, we modify the agent dynamics by adding stochasticity to dynamics which is unknown to the human.

- **True Dynamics:** The dynamics is a bit more stochastic by applying a small amount of noise to some of the transitions.
- **Human Belief:** The agent is operating reliably.

The agent's stochasticity would largely influence the agent's policy learning and may behave differently compared to when the agent is more reliable. Such discrepancy brings about the misguided preference data and would impact the reward learning with the implicit assumption that the human holds a correct belief about the agent's dynamics.

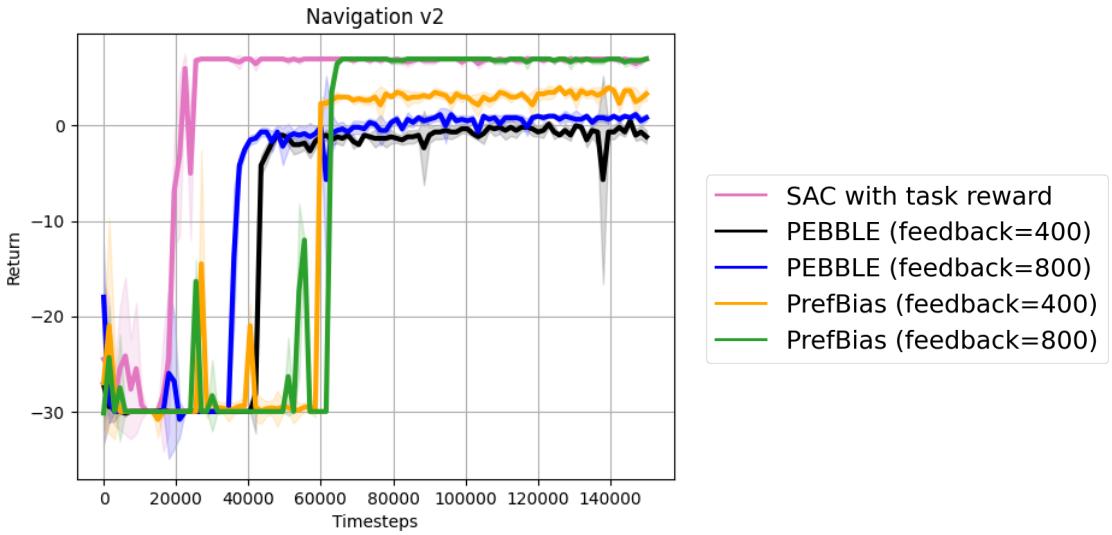


Figure 17. Learning curves on the task Navigation 2 as measured on the ground truth reward.

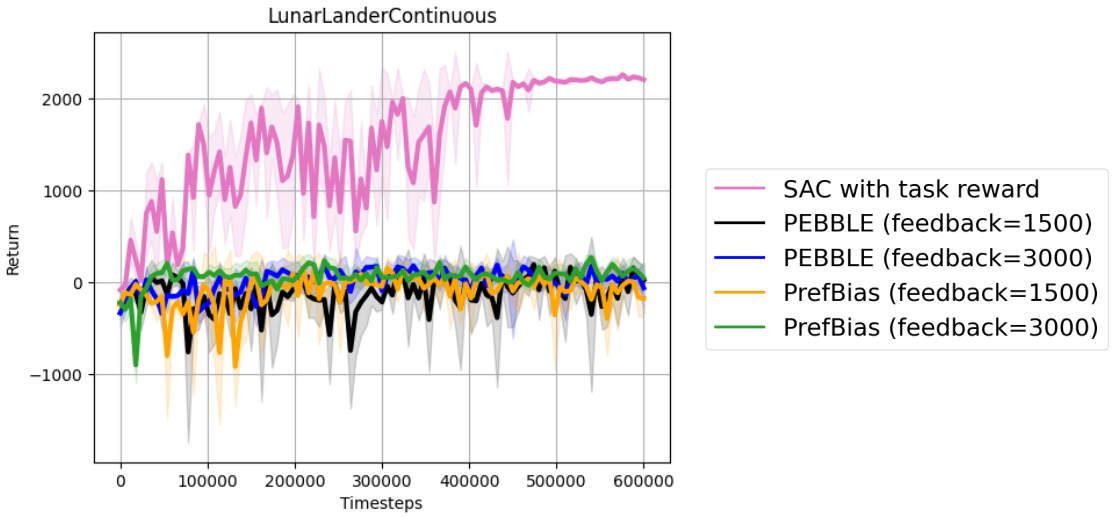


Figure 18. Learning curves on the lunar lander task as measured on the ground truth reward.

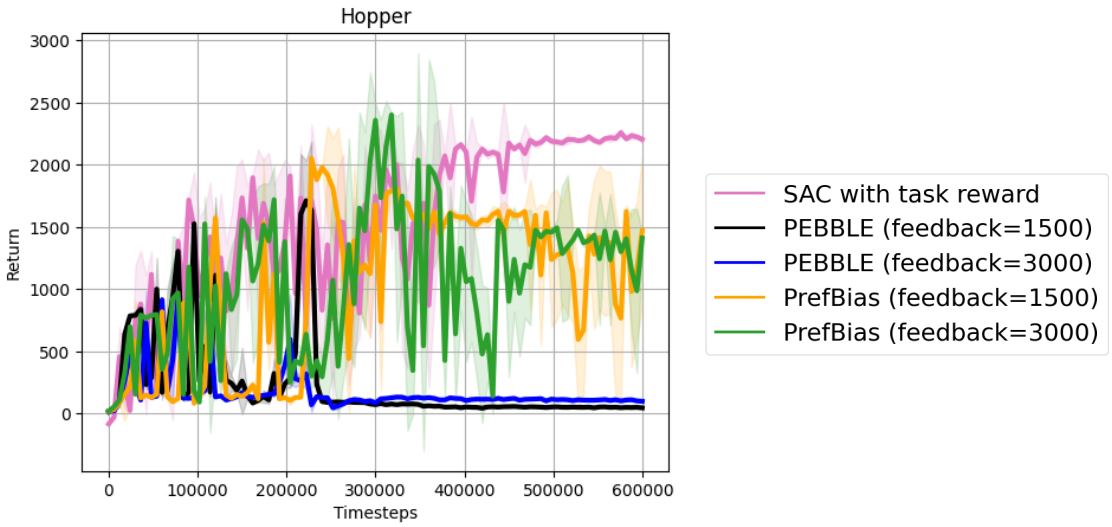


Figure 19. Learning curves on the hopper task as measured on the ground truth reward.

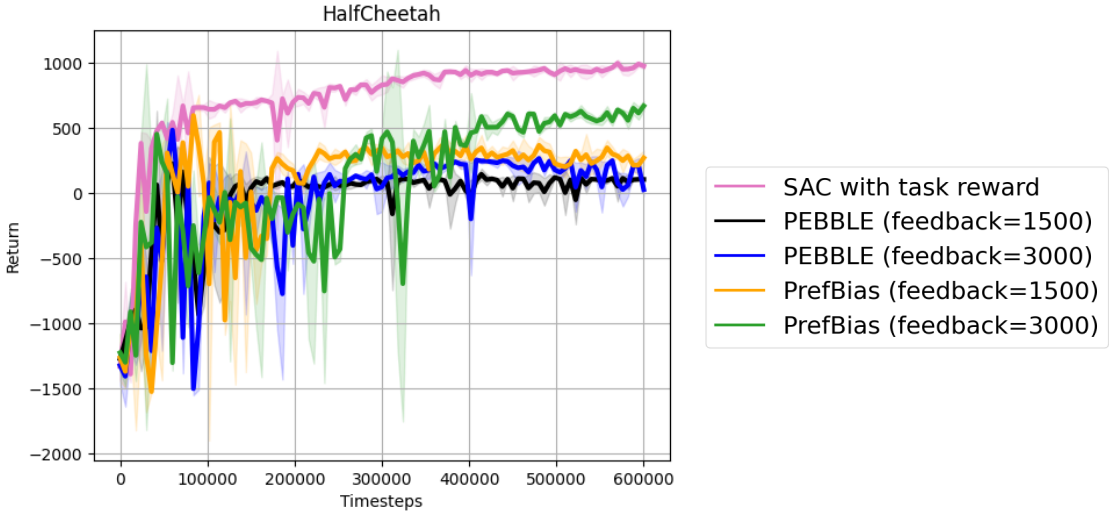
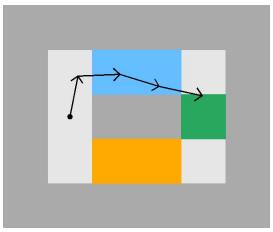
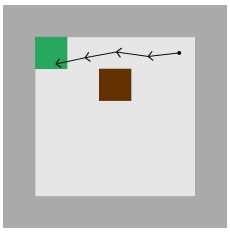


Figure 20. Learning curves on the cheetah task as measured on the ground truth reward.



(a) Navigation 1 (Left: PrefBias; Right: PEBBLE)



(b) Navigation 2 (Left: PrefBias; Right: PEBBLE)

Figure 21. Trajectories learned by PEBBLE and PrefBias agent in 2D navigation domains. For each domain, the left is for PrefBias agent while the right is for PEBBLE agent.

3.4.3 Results and Discussions

Figure 16, 17, 18, 19, and 20 shows the learning curves of PrefBias with different number of synthetic human feedback and that of PEBBLE on five testing domains. We also show the learning curve of SAC which serves as oracle performance. For domains Navigation 1 and Navigation 2 (see Figure 16 and 17), PrefBias (green) with 800 queries reaches the performance as SAC (pink) while PEBBLE obtains less return. We believe that this is because the surrogate reward function learned by PEBBLE is misguided by the biased belief and hardly capture the human's objectives. Especially for navigation 1, we notice that the trajectory returns of PEBBLE agent first reach the oracle performance and then goes down. That is because the PEBBLE agent initially finds the near-optimal policy during exploration, while the learned surrogate reward function later misleads the agent to navigation other areas while avoiding the icy corridor. Moreover, note that PrefBias would take more timesteps to converge since it has two models and more parameters to optimize and needs more data and timesteps to learn. The trajectories of PrefBias and PEBBLE agents is illustrated in Figure 21. For each domain, PrefBias is on the left is PrefBias and PEBBLE is on

the right. We can see that in navigation 1, the PEBBLE agent navigates through the sandy corridor which gives us a suboptimal behavior because it learns that it will be penalized navigating on ice. PrefBias agent could successfully neutralize the biased belief and recover the optimal behavior as SAC. Similarly, in navigation 2, PEBBLE learns to keep distance from the pit area and take a detour subject to the learned reward subject to the biased belief, while PrefBias could find the shortcut to the goal.

Figure 18, 19, 20 show the performance of PrefBias and PEBBLE on three more complex domains. For Lunar Lander (see Figure 18), both PrefBias and PEBBLE fail to match the SAC performance. For Hopper (see Figure 19), PEBBLE presents very poor performance as almost stay around a return of 0 after attempts at the early stage of learning process while PrefBias achieves much better performance. But notice that PrefBias has larger variance since the Prefbias may take more data and steps to learn. Similarly, for Cheetah (see Figure 20), PrefBias can better match the performance of SAC, compared to PEBBLE baseline.

3.4.3.1 Effects of Different Priors

One problem of previous PbRL methods is that they omits the effects of potential biased belief and only learn a surrogate reward function which would be subject to the biases. Thus, it is important to consider human belief while learning reward from the human preferences. We believe that a better understanding of the human belief could contribute to learning a reward function that describes what the human really wants. Build on a BNN framework, we could leverage the advantage of prior to infer

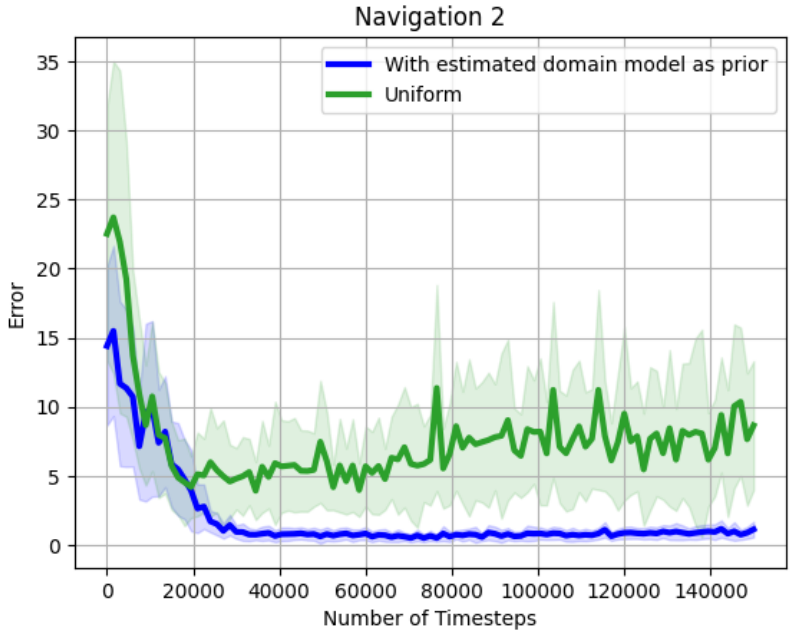


Figure 22. L2 error between estimated human belief model and ground truth on the navigation task.

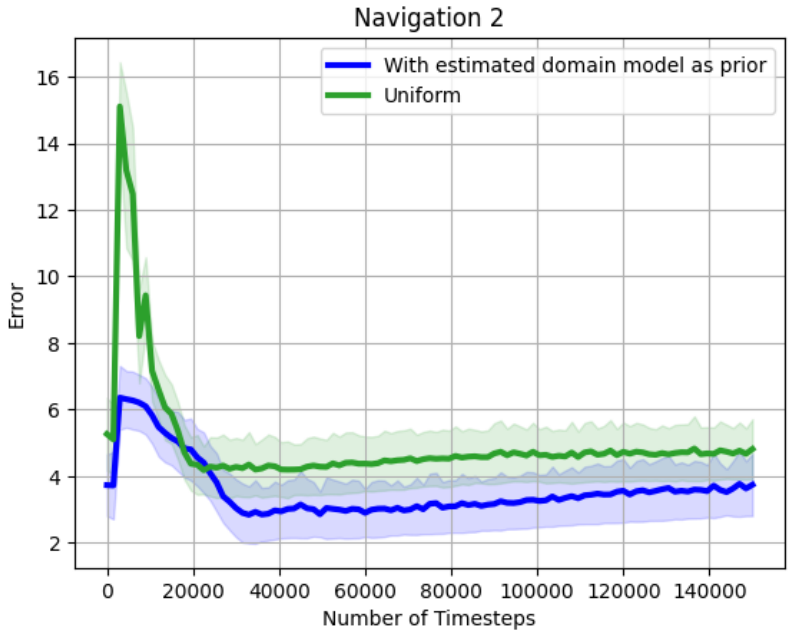


Figure 23. L2 error between surrogate reward function and ground truth on the navigation task.

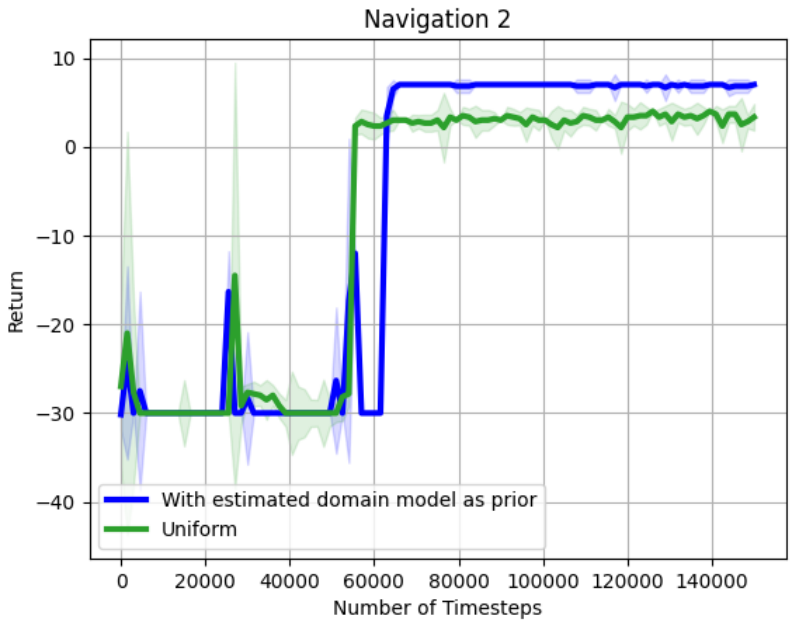


Figure 24. Learning curves on the navigation task as measured on the ground truth reward.

the human belief. In this work, we make the prior to be a Gaussian distribution, but explore two ways to set the mean of it: Random initialization and True Domain Dynamics-based initialization. For the first method, we randomly configure the mean of the prior distribution. For the second method, we assume that the human belief is a disturbed true domain dynamics and use the parameters of an estimated true domain dynamics model as the mean of the prior distribution. Figure 22 shows the comparison between these two different priors in terms of human belief learning. We can see that PrefBias can learn a belief that matchs the ground truth while taking the parameters of an estimated domain dynamics model as the prior. Similarly, the True Domain Dynamics-based initialization outperforms the random initialization in terms of reward and policy learning (see Figure 23 and 24). Therefore, taking the true domain dynamics into consideration while setting the prior could help us estimate the human belief model and learn a surrogate reward function that better captures the

human's objectives. Moreover, we believe that a prior that contains information of the model differences and domain knowledge would further benefit the learning process.

3.4.3.2 Explore Different Query Sampling Methods

In the experiments, we use diagreement-based sampling method for PEBBLE which gives us best performance. For PrefBias, we investigate the performance of several query sampling methods: Uniform, Disagreement-based, Entropy-based sampling. We also introduce an alternative sampling method, especially for our problem scenarios. We call it Preference Disagreement-based sampling. It aims to select queries that maximize the difference of the preference estimated in terms of agent's and human model which could help us infer the biases in the human belief efficiently. We tested all the sampling methods. The preference disagreement-based sampling demonstrates slightly better performance, but not much. That could be because the queries selected elicit information much about the belief, but not to balance the information gain of the reward learning.

3.5 Conclusion

In this work, we investigate the problem of PbRL under human biased belief. Without consideration of the human belief, the learned surrogate reward function could be wrong and result in suboptimal or unintended behaviors. We propose to learn a reward function while taking the human belief into account. Both reward function and human belief are modeled using BNN. Their parameters are updated from the human preference data in a variational Bayesian framework. The learned

reward function is then used to optimize the agent's policy. We evaluate and compare our method with a state-of-the-art PbRL baseline on several continuous control tasks. The results show that our method can successfully neutralize the belief biases and reach near-oracle performance compared to the baseline method.

Note that we focus on the scenarios that the human is not observing the agent's policy execution where the agent should work in a effective way with respect to its own model and optimize the true reward functions. In such cases, the agent may behavior weirdly from the human's perspective, but is effective with respect to the true dynamics. There raises the problem of how to get an explicable behavior to the human. It will be investigated in Chapter 4.

Chapter 4

EXPLICABLE POLICY SEARCH

4.1 Introduction

Intelligent agents are quickly becoming a part of our daily lives in a variety of domains, including smart home, autonomous driving, education, and so on. In such domains, the agents are expected to perform in human inhabited environments and even collaborate closely with them. Members in such a collaborative setting often form conscious and subconscious expectations of each other and the success of collaboration depends on whether such expectations can be met. A key challenge here is that the human's expectation may not align with the agent's optimal behavior. Hence, agents choosing their optimal behaviors without considering the human observers or collaborators could be seen as unexpected, leading to degraded team performance and loss of trust (Gunning 2017; Chakraborti, Kambhampati, et al. 2017). To be good team players, these agents are required to respect human's expectation for their behaviors.

Consider a drone navigation scenario in Figure 25 where a drone is tasked to navigate from the starting position (as shown) to a destination. On a good day, the drone would be expected to mostly navigate straight to the goal with some air perturbation. However, there is a heavy wind today that changes the domain dynamics (which a human observer is unaware of) and it becomes impossible for the drone to navigate as usual. Two alternatives are illustrated where the drone navigates in a zigzag and a curved pathway. The curved pathway is more optimal since it

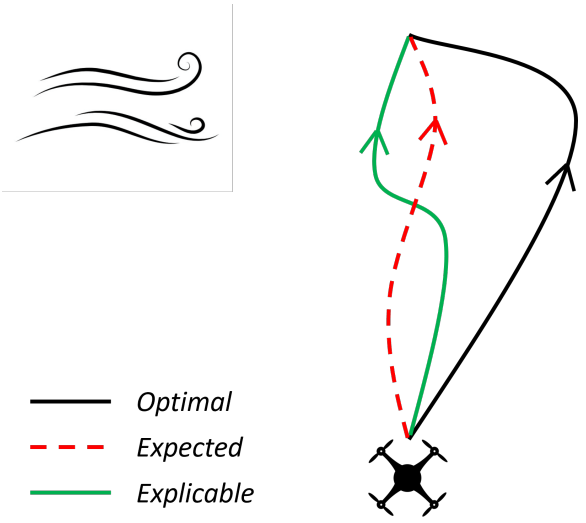


Figure 25. Motivating scenario to demonstrate explicable trajectories.

makes fewer sharp turns (due to the windy condition) but the former is likely to be perceived as more expected to the human observer, given its *closeness* to the expected behavior. We note here that the explicable behavior goes beyond simply choosing from or combining the agent's optimal behavior with the human's expected behavior. Hence, achieving such a capability requires a more fundamental treatment. In the experiments, we will show that our learning agent can generate novel behaviors that are rarely seen in training scenarios.

The problem setting of *explicable policy search (EPS)* is illustrated in Fig. 26. The learning agent learns its behavior with a given engineered reward model, r_A, in the task environment under the true domain dynamics $\mathcal{T}_A$. The human generates her expectation of the agent's behavior π_A^H based on $\mathcal{T}_A^H$ that captures her belief or understanding of the true domain dynamics, and her reward model r_H, which may be different from $\mathcal{T}_A$ and r_A, respectively. The differences may be due to personal preferences in the reward model, biases and misunderstandings about the domain

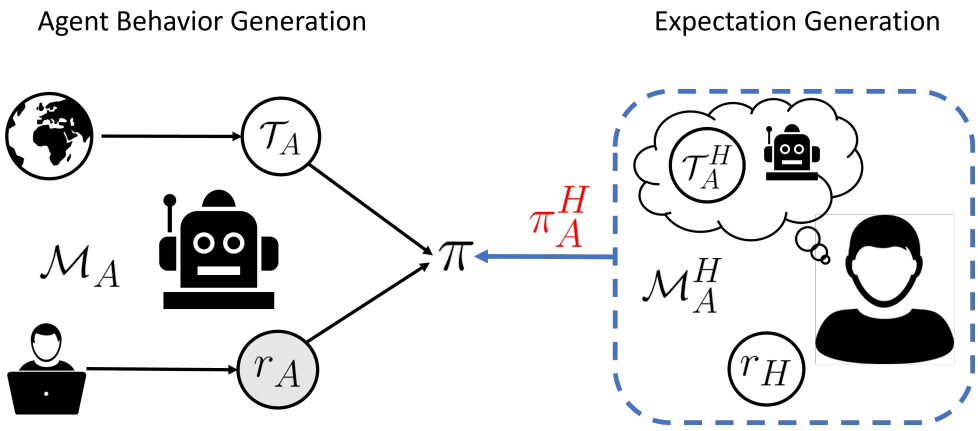


Figure 26. Problem setting of EPS. Shaded nodes are known to the agent and unshaded nodes are unknown.

dynamics, etc. In the traditional learning setting, the agent computes the optimal policy π under its models for behavior generation, which could be different from its expectation formed by the human based on her models. Note that r_A and r_H could be different since r_A may not accurately describe the human's preferences. In this work, we encourage the agent to respect $\mathcal{T}_A^H$ since: (a) it is hard to change; (b) the agent's behavior may become inexplicable and even lose human trust if we don't take $\mathcal{T}_A^H$ into consideration. Moreover, the agent is tasked to consider both r_A and r_H instead of r_H merely, for two reasons: (a) r_A potentially captures the human's preference to some extent and it could make the learning process faster as an inductive bias; (b) r_A may specify some attributes regarding the agent's ability by the system designer which are neglected or unknown to the human.

EPS is fundamentally a model reconciliation business (Chakraborti et al. 2019). In such a setting, considering the learning problem only from the perspective of the agent would sabotage the teaming when the agent's behavior differs from human's expectation. Our method considers any feedback the human may provide to the agent based on her expectation of its behavior (generated under her models) to help the

agent improve. Hence, our method has a variety of applications where machines must be tuned to individual users, such as an autonomous vehicle that learns to improve its driving behavior based on its owner's feedback but at the same time must abide by standardized operating requirements to ensure safety, comfort, etc. In our problem, we assume that the human is nosily rational at generating her expectation of the agent to accommodate her computational limitations. While noisy rationality may not perfectly describe our computational model (Shah et al. 2019) due to various cognitive biases, modeling humans as such has been a common practice and demonstrated as a reasonable assumption in prior methods (Oaksford and Chater 2007; Baker, Saxe, and Tenenbaum 2011). Furthermore, similar to prior work on explicable planning (Zhang et al. 2017; Kulkarni et al. 2019), we assume that the human is a sole observer of the agent. Extending it to an collaborative setting requires additional machinery and will be discussed in future work, with steps already taken in a classical planning setting (Zakershahrak et al. 2018).

In this work, we formulate *explicable policy search (EPS)* in Markov Decision Processes (MDPs). The goal is to learn a policy that reconciles between maximizing the long-term return (based on the engineered reward model r_A) and minimizing the deviation of the agent's behavior from the human's expectation of it, which is quantified by an *explicability* score. We formulate the problem as a linear combination of two objectives and show that such a simple treatment already yields substantial benefits in our evaluation. The challenge in this problem formulation mainly lies in the fact that the human's expectation is hidden and must be learned. A straightforward solution involves learning both the human's belief about the domain dynamics ($\mathcal{T}_A^H$) and her reward model (r_H) from human feedback, which is possible but impractical. Instead, we show that the information needed from them for EPS can be sufficiently

encoded by a surrogate reward function, which is much easier to learn. Such a reward function can then be incorporated into a policy search process to recover an explicable policy. The final modified reward function for EPS bears some similarity to maximum entropy reinforcement learning (Haarnoja et al. 2017; Haarnoja, Zhou, Abbeel, et al. 2018) but is derived under a completely different motivation.

For evaluation, we design a set of navigation domains where the human's feedback on her expectation of the agent's behavior is synthetically generated based on the true domain dynamics modified with various "misunderstandings". We compare our method with three baselines that are selected to represent the traditional RL methods to demonstrate the key advantages of EPS under model differences. Furthermore, to show its real-world relevance, we conduct a human study in an autonomous driving domain (Leurent 2018) where we design driving scenarios to elicit existing human biases about the vehicle's domain dynamics that could be dangerous to ignore during learning. The results show that our method can intelligently generate explicable behaviors that are safe and preferred over those of the baselines.

The contribution of this work is three-fold. First, we introduce and formulate explicable policy search, which extends explicable planning to a reinforcement learning setting and to stochastic domains with continuous state and action spaces. Second, we propose a practical solution for this challenging problem by introducing a surrogate reward function learned from human feedback data, which sufficiently encodes the necessary information for explicable policy search. Third, we evaluate our learning method extensively with simulations and human subjects.

4.2 Related Work

The problem of generating communicative actions or behaviors (a.k.a. explainable planning) has been well studied as a subarea in explainable AI. Various terms have been introduced for explainable planning that are different but share similarities, such as legibility, explicability, transparency, etc. For a review of their differences, refer to here (Chakraborti et al. 2019). For example, legible motion planning (Dragan, Lee, and Srinivasa 2013) is about generating motion trajectories to better reveal the underlying goal of the agent. Explicable planning that we study in this work (Zhang et al. 2017; Kulkarni et al. 2019; Gong and Zhang 2018; Zakershahrak et al. 2018) differs from legible planning in that it focuses on plans that better align with the human's expectation *given the goal*. The key characterization of explicable planning methods revolves around the idea of model reconciliation where an agent makes decisions based on two different models instead of one (with a focus on domain dynamics) (Chakraborti, Sreedharan, et al. 2017; Chakraborti, Kambhampati, et al. 2017; Chakraborti et al. 2019). Zhang et al. (Zhang et al. 2017) formulated the problem as a learning and planning problem, where the human's expectation of the agent's behavior is learned through a labeling process. A metric for explicability is defined based on the learned labeling schema and then used to regularize the planning process to synthesize explicable plans. Kulkarni et al. (Kulkarni et al. 2019) considered it directly as a distance learning problem (Chakraborti, Sreedharan, et al. 2017) and generated explicable plans by minimizing an explicability distance between plans from the two models. A strong assumption was made that the human's model was provided a priori. while it is difficult to obtain in most cases. Prior methods on explicable planning addressed the problem in a classical planning setting (e.g.,

PDDL (Fox and Long 2003)) under deterministic domains. We argue that explicable planning should also be considered in a learning setting where human users can provide feedback on the agent's behavior for the agent to improve over time (i.e., for personalization). Furthermore, extending explicable planning to a learning setting makes the approach applicable in stochastic environments with continuous state and action spaces. Differing from the prior work, we consider that the user's reward model may also differ from the agent's, but assume the differences do not introduce a discrepancy in the perception of the agent's "goal". Considering multiple candidate goals is the setting of legible planning (Dragan, Lee, and Srinivasa 2013).

As seen from Figure 26, estimating the human's expectation requires learning both the human's belief about domain dynamics and her reward model based on human feedback. Existing work on inverse reinforcement learning (IRL) (Abbeel and Ng 2004; Ramachandran and Amir 2007; Ziebart et al. 2008) and reward learning (Daniel et al. 2014; Sadigh et al. 2017; Erdem et al. 2020) learns the reward model from human data with the assumption that the human has access to an accurate model of domain dynamics. However, they implicitly assume that the human maintains an accurate understanding of the agent's dynamics and explain any deviations from optimality as noise. While noise introduces variations, bias determines the average of errors (Kahneman 2011). Human biases can significantly impact our decisions and judgments. It was shown that the human's inaccurate belief of domain dynamics may skew the human feedback and lead to learning the opposite preferences with respect to the human's true reward model (Gong and Zhang 2020). A similar situation exists in policy learning for which various methods have been devised that use human feedback, which include reward shaping, policy shaping (Griffith et al. 2013), and interactive RL (Knox and Stone 2009; MacGlashan et al. 2017; Christiano et al. 2017; Lee, Smith,

and Abbeel 2021). When the human has an inaccurate belief of domain dynamics, as we will show, it can lead to non-convergence during learning or high variances in the learned behaviors for these methods. While it may be possible to learn the human's belief of the domain dynamics separately (Reddy, Dragan, and Levine 2018), the belief and human's reward model are generally tightly coupled in the human feedback and should be jointly learned. Joint learning is possible (Herman et al. 2016; Gong and Zhang 2020) but very challenging due to its high dimensionality, which makes it impractical for real-world domains. When considering only differences in domain dynamics, our problem may be viewed as a form of off-dynamics learning (Eysenbach et al. 2020) that addresses transfer learning from a source to a target domain. However, we do not have direct access to the target domain in EPS, which is hidden in the human's mind.

4.3 Our Approach

4.3.1 Problem Formulation

In this work, we formulate any task domain as a Markov Decision Process (MDP). An MDP is represented by a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, r, \rho, \gamma)$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ the set of actions, $\mathcal{T}(s|s, a)$ the transition function, r the reward function, $\rho(\cdot)$ the initial state distribution, and γ the discount factor. For the problem setting as shown in Figure 26, we need to consider two MDPs. Assuming the two MDPs share the same $\mathcal{S}$, $\mathcal{A}$, ρ, and γ, one is the agent's model $\mathcal{M}_A$ with the true domain dynamics $\mathcal{T}_A$ and the engineered reward function r_A, and the other is the human's model of

expectation $\mathcal{M}_A^H$ with her belief about the domain dynamics $\mathcal{T}_A^H$ and the human's reward function r_H. For explicable policy search, $\mathcal{T}_A$, $\mathcal{T}_A^H$, and r_H are *unknown*.

Definition 1. Explicable Policy Search (EPS) *is the problem of searching for a policy via learning to maximize two objectives: the expected cumulative reward, and a* ***policy explicability score*** *between the agent's policy under $\mathcal{M}_A$ and the expected policy under the human's model $\mathcal{M}_A^H$.*

In this work, we consider a linearly weighted sum of the two objectives:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi,\mathcal{T}_A}\underbrace{\left[\sum_t \gamma^t r_A(s_t, a_t)\right]}_{cumulative\ reward} + \lambda \underbrace{\mathcal{E}(\pi, \mathcal{M}_A, \pi^H, \mathcal{M}_A^H)}_{policy\ explicability\ score}, \tag{4.1}$$

where π and π^H denote the agent's policy and human's expected policy, respectively. We combine the two objectives linearly via a *reconciliation factor* λ to be consistent with the literature (Zhang et al. 2017; Kulkarni et al. 2019) and simplify the technical development.

We introduce the *policy explicability score* for stochastic environments, which differs fundamentally from the explicability scores defined in the prior work. The policy explicability score does not consider the similarity between any two trajectories as considered in the explicability scores in the prior work: they are *orthogonal* aspects of explicability that are equally important. In (Zhang et al. 2017; Kulkarni et al. 2019), explicability scores have been considered as a similarity metric between the agent's plan and the expected plan in the human's mind. Working in a stochastic setting, foremost, requires us to consider distributions of trajectories. Hence, a natural choice for the policy explicability score is the negative KL-divergence of the two distributions of trajectories under the agent and human's models and policies, respectively. In such a case, the second objective in Equation (4.1) becomes a measurement of consistency between the generated agent's trajectories and the human's expectation. Intuitively,

this means that an agent that only learns to maximize it would learn to *replicate* (as closely as possible) what the human expects the agent to behave–exactly what we set out to pursue! In such a case, Equation (4.1) can be rewritten as:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi,\mathcal{T}_A}\left[\sum_t \gamma^t r_A(s_t, a_t)\right] + \lambda \cdot -\mathcal{D}_{\mathrm{KL}}(p_A(\tau) \| p_A^H(\tau)), \tag{4.2}$$

where $p_A(\tau)$ and $p_A^H(\tau)$ denote the distributions of trajectories for the agent and human, respectively.

Remark: Depending on whether the agent's state and action are observable, the distribution of trajectories must be computed differently. When both the state and action are observable, the human would be able to contrast both the agent's action and resulting state with her expectation; otherwise, only the observable part needs to be considered. In the following discussion, we assume that both the state and action are observable. For explicable policy search, the implication here is that the agent optimizing Equation (4.2) would actively avoid parts of the state space where either its policy differs from the human's expectation or the dynamics differs, which is reflected in the final solution derived.

4.3.2 Explicable Policy Search (EPS)

To present our learning method, we start by expressing the probability distribution of the agent's trajectories and human's expectation. We parameterize the agent's policy using θ. The probability of realizing the agent's trajectory τ with π_θ (and similarly for the human's expectation) is:

$$p_A(\tau) = \rho(s_0) \prod_t \mathcal{T}_A(s_{t+1}|s_t, a_t)\pi_\theta(a_t|s_t), \quad p_A^H(\tau) = \rho(s_0) \prod_t \mathcal{T}_A^H(s_{t+1}|s_t, a_t)\pi_A^H(a_t|s_t). \tag{4.3}$$

Given these two distributions, we can now derive our solution for EPS. Since both objectives in Equation (4.2) are expressed as expectations over the same distribution of trajectories (i.e., $p_A(\tau)$), we can combine them after rewriting the policy explicability score in Equation (4.2) as follows:

$$\begin{aligned} & -\mathcal{D}_{\mathrm{KL}}(p_A(\tau)\|p_A^H(\tau)) \\ &= -\mathbb{E}_{p_A}\left[\sum_t \log \mathcal{T}_A(s_{t+1}|s_t,a_t)+\log \pi_\theta(a_t|s_t)-\log \mathcal{T}_A^H(s_{t+1}|s_t,a_t)-\log \pi_A^H(a_t|s_t)\right]+C, \end{aligned} \tag{4.4}$$

where C is a constant. The main challenge in solving the optimization problem in Equation (4.2) now lies in the fact that neither the human's policy π_H nor her belief of the domain dynamics $\mathcal{T}_A^H$ are known.

4.3.3 Surrogate Reward Function

Equation (4.4) can be merged into Equation (4.2) such that the log terms essentially reshape the engineered reward function. In such a case, a straightforward approach is to learn the human's belief about the domain dynamics and her expected policy separately based on human feedback, while maintaining estimates of the true domain dynamics and current agent's policy. While possible, it is inefficient and unnecessary. Instead, we propose to use a surrogate reward function. The goal is to learn such a function u_H that retains the necessary information about the human's belief of domain dynamics and expected policy for policy search, i.e., $u_H \doteq \log \mathcal{T}_A^H(s_{t+1}|s_t,a_t) + \log \pi_A^H(a_t|s_t)$.

Intuitively, we learn a reward function that alone can explain the human's expectation of the trajectories–a reward function that introduces the same distribution of the expected trajectories. At the same time, we must take care to minimize the

human feedback to make it practical. A popular approach that imposes feasible input requirement on human is preference-based RL (Wirth et al. 2017; Christiano et al. 2017). For EPS, instead of soliciting human preferred trajectories, we can present pairs of trajectories and ask humans to comment on which one is *more expected*. Then, we can fit a reward function that would "prefer" more expected trajectories. More specifically, we can try to correlate the distribution of expected trajectories with the surrogate reward function to be learned as follows:

$$p_A^H(\tau) \propto \exp\left(\sum_t u_H(s_t, a_t)\right). \tag{4.5}$$

Proposition 1 *There exists a* ***unique*** *reward function u_H such that trajectory distribution under the softmax human preference model with u_H described above matches with the human's expected trajectory distribution given in Equation* (4.3).

When we compare mathematically the two equations of $p_A^H(\tau)$ (i.e., Eqs. (4.3) and (4.5)), we see that the only way to match the distributions is by satisfying:

$$\sum_t u_H(s_t, a_t) = \sum_t \log \mathcal{T}_A^H(s_{t+1}|s_t, a_t) + \sum_t \log \pi_A^H(a_t|s_t) + C_1.$$

One straightforward way for this is to set $u_H(s_t, a_t) = \log \mathcal{T}_A^H(s_{t+1}|s_t, a_t) + \log \pi_A^H(a_t|s_t)$. In the other direction, since the trajectories may be of various lengths in general (including the length of 1 as special cases), we can conclude further that:

$$u_H(s_t, a_t) = \log \mathcal{T}_A^H(s_{t+1}|s_t, a_t) + \log \pi_A^H(a_t|s_t) + C_1.$$

Plugging this result into any trajectory of length greater than 1, we can conclude that $C_1 = 0$.

This means that u_H in such a case becomes equivalent to $\log \mathcal{T}_A^H$ and $\log \pi_H$ for representing the distribution of expected trajectories, which implies that the

information needed from $\mathcal{T}_A^H$ and π_H for optimizing Equation (4.2) can be substituted with u_H. u_H can be learned by applying preference-based learning with a softmax preference model based on human feedback as described earlier (more details to follow). To learn the unique surrogate reward function and avoid the non-identifiabillity issue (Russell 1998), however, requires the learning to be globally optimized. In our implementation, we simply normalize the learned rewards. Now, we rewrite Equation (4.2) based on the above result:

$$\theta^* \doteq \arg\max_{\pi_\theta} \mathbb{E}_{p_A}\left[\sum_t \gamma^t \Big(r_A(s_t, a_t) + \lambda\big(u_H(s_t, a_t) + \mathcal{H}_{\mathcal{T}_A}[s_{t+1}|s_t, a_t] + \mathcal{H}_{\pi_\theta}[a_t|s_t]\big)\Big)\right]. \tag{4.6}$$

Note that Equation (4.6) is an approximation of the original objective in Equation (4.2) by ignoring the influence of the discount factor on the surrogate reward function and entropy terms. This allows us to use them to reshape the reward function. We can view this objective function as two parts. The first two terms, $r_A(s_t, a_t) + \lambda u_H(s_t, a_t)$, requires the agent to maximize rewards from two sources: the engineered reward and the surrogate reward learned from human feedback. These two reward functions are weighted according to the reconciliation parameter as expected. The second part of this objective function is $\mathcal{H}_{\mathcal{T}_A}[s_{t+1}|s_t, a_t] + \mathcal{H}_{\pi_\theta}[a_t|s_t]$, which are two entropy terms. The first term is for the agent's domain dynamics while the second term is for the target policy.

The second entropy term for the target policy (referred to as "*policy entropy*") is well studied in the maximum entropy (MaxEnt) RL framework. (Haarnoja et al. 2017; Haarnoja, Zhou, Abbeel, et al. 2018), However, we note that this term in our work is derived for a completely different reason. In MaxEnt RL, maximizing $\mathcal{H}_{\pi_\theta}$ encourages the agent to explore during learning and increase robustness. Given a fixed surrogate reward function, this term in EPS likewise encourages stochasticity in the agent's

policy to increase robustness. A more distinguishing feature of explicable policy search lies in the first term. Maximizing $\mathcal{H}_{\mathcal{T}_A}$ encourages the agent to prefer parts of the environment where there is more stochasticity, which we refer to as "*environment entropy*". The incentive here is to reduce the influence to the policy explicability score due to differences in the agent's policy and expected policy. Intuitively, at more stochastic parts of the environment where action choices for the agent matter less, the agent's trajectory is more likely to match with the human's expectation even when the agent's policy and the expected policy differ. Exploring in stochastic parts of the environment would provide more flexibility to search for the behavior that is explicable to the human.

Connections to RL problems: The optimization target in Equation (4.6) relates to those used in various RL methods under special conditions. For example, when the underlying domain is deterministic, the optimization criterion becomes that of a multi-objective maxEnt RL problem. When r_A is identical to u_H, the optimization criterion is exactly that of off-dynamics RL. When the human maintains an accurate belief about domain dynamics and her reward function coincides with the design reward function, it reduces to standard RL. Moreover, although our formulation presents such close connections, the main difference from others is that we aim to reconcile the task performance with explicability level in terms of human model.

Expectation-based Preferences: For learning u_H, we apply a preference-based learning framework. We request humans to provide their feedback on which segment in a pair of segments $\{(\sigma^1, \sigma^2)\}$ extracted from the agent's trajectories *is more expected*. To consider noise in human feedback, the human's expectation preference is formulated as follows (Christiano et al. 2017):

$$\hat{P}[\sigma^1 \succ \sigma^2] = \frac{\exp \sum u_H(s_t^1, a_t^1)}{\exp \sum u_H(s_t^1, a_t^1) + \exp \sum u_H(s_t^2, a_t^2)}$$

Algorithm 3 Explicable Policy Search (EPS)

Initialize the agent's policy π_θ, true domain dynamics $\mathcal{T}_A$, surrogate reward function u_H, and an empty set D.
for $t = 1 \cdots max$ iterations **do**
  Collect samples from environment using π_θ and add them to D.
  Train $\mathcal{T}_A$ using D.
  repeat
    Select m pairs of trajectory segments from D to solicit expectation preferences.
    Learn u_H using human feedback data.
  until presented a predefined number of sample pairs
  if reached a predefined batch size **then**
    Update π_θ based on the reshaped reward in Equation (4.6).
  end if
end for
return π_θ

We learn u_H to minimize the cross entropy between our prediction of the expectation preferences and feedback data. In order to efficiently solicit the human's expectation preference for the agent's behavior, we leverage uncertainty-based sampling (Christiano et al. 2017; Lee et al. 2021) to select traces for the human in an active learning fashion. To estimate $\mathcal{T}_A$, we assume it follows a Gaussian distributions and model it using a two-headed neural network that outputs its mean and logarithm of standard deviation given the state and action. The agent interacts with the environment and collects transition data for learning. $\mathcal{T}_A$ is estimated using a data-driven method by minimizing the L_2 one-step prediction loss (Kurutach et al. 2018). We use Soft Actor Critic (SAC) (Haarnoja, Zhou, Abbeel, et al. 2018) for policy learning but other policy search methods are also applicable. To alleviate the issue of non-stationary prediction of expectation preferences during learning, we relabel the data samples every time u_H is updated (Lee, Smith, and Abbeel 2021). The algorithm for EPS is present in Algorithm 3.

4.4 Evaluation

We evaluate our method on a set of continuous navigation domains with synthetic human models and a simulated autonomous driving domain with a human subject study. The study is IRB approved and all protocols have been followed. The synthetic experiments are used to validate the effectiveness of our method for searching for explicable policies. The user study is to 1) confirm that our beliefs about domain dynamics can be inaccurate or biased, which can affect our judgements and lead to severe consequences if ignored, and 2) show that our method can effectively address such hidden issues by achieving an intelligent reconciliation between the task performance and human's expectation.

4.4.1 Synthetic Navigation Domains

We conduct experiments on four navigation domains with continuous state and action spaces, as illustrated in Figure 27. The environment of all the task domains is in the form of a 7×7 continuous grid-world. The state space consists of the 2D location of the agent in the domain. The action space consists of velocity and angle of the agent's move. Moreover, we added Gaussian noises to each move to simulate stochasticity in the real-world. For all the domains, the agent starts from the upper left corner and aims to navigate to the goal area (depicted in green). We introduce the domains as follows:

- **Domain 1 (D1):** This domain is adapted from the classical cliff walking domain (Sutton and Barto 2018). There is one pit area with -100 penalty and a goal area with $+100$ reward in the environment. The reward of each (s, a) pair

depends on how much the action a forwards the agent towards the goal when it is at state s. Moreover, there is an additional living reward (i.e., -1) for each step. The environment is shown in the first row in Figure 27. The dark grey area around represents the walls. The brown block is the pit and the green block in the upper right corner is the goal. The agent starts from the upper left corner and aims to navigate to the goal.

- **Domain 2 (D2):** Similar to domain 1, this domain contains one pit area and a goal area. The true rewards and dynamics are the same as domain 1. The only difference is that the location of the pit area is moved one block down. Now, the agent has two possible ways to reach the goal that are separated by the pit. The environment is shown in the second row in Figure 27.
- **Domain 3 (D3):** As shown in the third row in Figure 27, the goal is at the bottom right corner of the environment. There are two paths starting from the upper left corner (i.e., the initial position) to the goal separated by an impassable area in the middle. The path passing through the top is an icy road (depicted in blue) and the pit is at its right end. The other path passing through the left is covered with sands (shown in yellow). The environment reward is the same as the first two domains. However, the dynamics model is different while navigating on different road conditions. In this domain, the agent is adept at moving on the icy road while extremely clumsy (i.e., can easily get stuck) on sand.
- **Domain 4 (D4):** The domain is a modified version of domain 3. The only difference is that the agent is now more proficient with sands–it is slower on sand but still maneuverable. The environment is shown in the fourth row in

Figure 27 where the sandy road on the left is illustrated in darker yellow to indicate that it is more navigable for the agent.

We built in various human biased models for these four domains. For simplicity, we modified the human's belief about the domain dynamics only while keeping the reward function the same. This should not impact the evaluation results since we have shown that the surrogate reward function can encode biases in both. For Domain 1 (D1) and Domain 2 (D2), the human believes that the agent is more likely to fall into the pit when it is close by (i.e., more stochastic) and the agent can navigate stably while further way; the truth is everywhere is the same. For Domain 3 (D3) and Domain 4 (D4), the human believes that the agent would easily slip on ice and it is more likely to fall into the pit while moving close by. Moreover, the human believes that the agent can readily handle sandy roads. The truth is that the agent is proficient on ice but can easily get stuck on sand in D3. For D4, the agent is in addition capable of handling sand albeit being more costly. In general, D1 and D2 is designed to demonstrate that EPS is able to generate novel behaviors other than the optimal agent's policy and the human's expectation. D3 and D4 examines how the EPS agent determines as it has to choose to follow between optimal agent's policy and the human's expectation. The human's preferences for the agent's behaviors is generated synthetically with respect to ground truth human model.

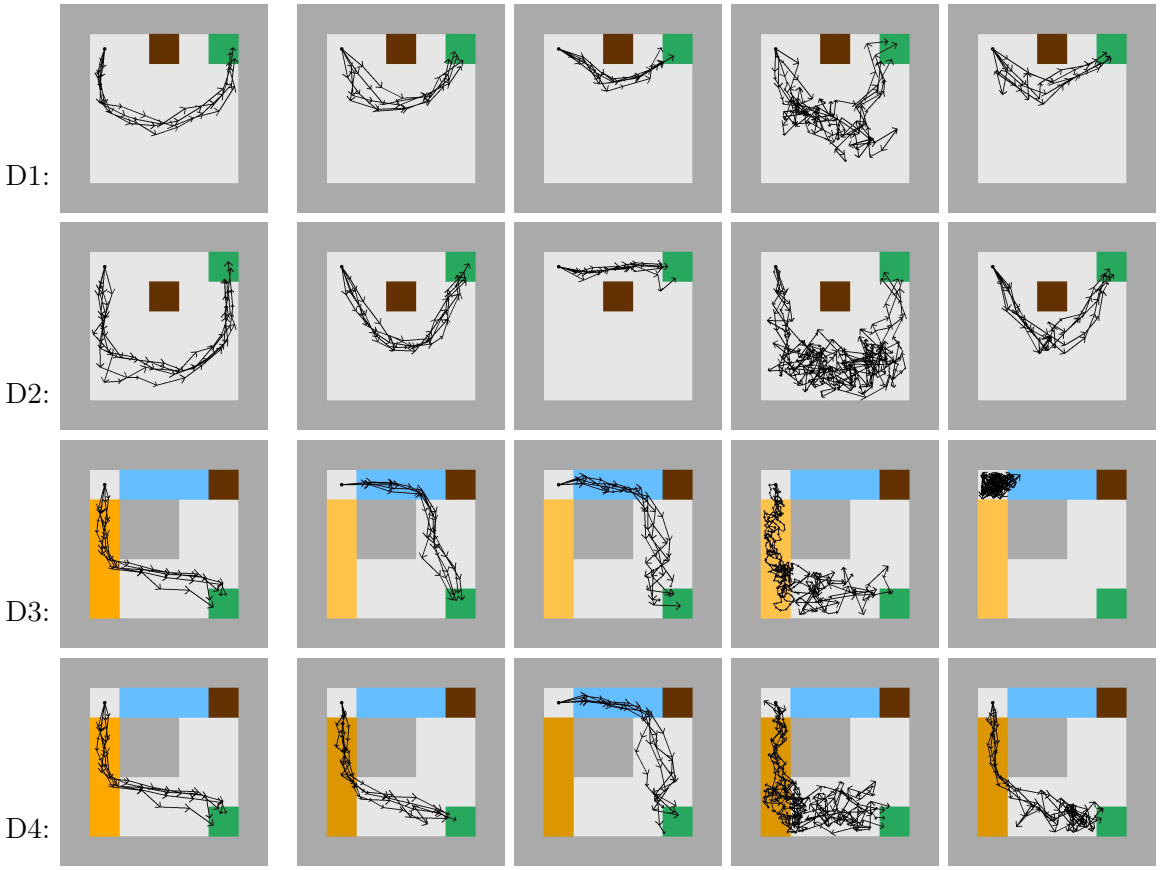
4.4.1.1 Baselines

We compared our method to three baselines to illustrate its advantages compared to the traditional RL methods that do not consider model differences: Soft Actor Critic (SAC) (Haarnoja, Zhou, Abbeel, et al. 2018), Deep RL from Human Preferences

(DRLHP) (Christiano et al. 2017), and Policy Shaping (PS) (Griffith et al. 2013). SAC optimizes policy with respect to the engineered reward function and true domain dynamics without considering the human's expectation. DRLHP uses human expectation-based preferences for the agent's behavior to estimate a reward function and applies it to guide policy search. It ignores the engineered reward function. For SAC and DRLHP, we could not use reward signals from both sources due to their different formats. The closest competitor to EPS is policy shaping. Policy shaping learns two policies using reward signals from the environment and human expectation preferences, respectively. A combination policy is then obtained by multiplying them together. It also seeks to combine different information sources that are however assumed to be consistent. That is why it has difficulty handling situations where the human's expectation and agent's optimal behavior conflict (see D3 in Figure 27).

4.4.1.2 Results and Discussion

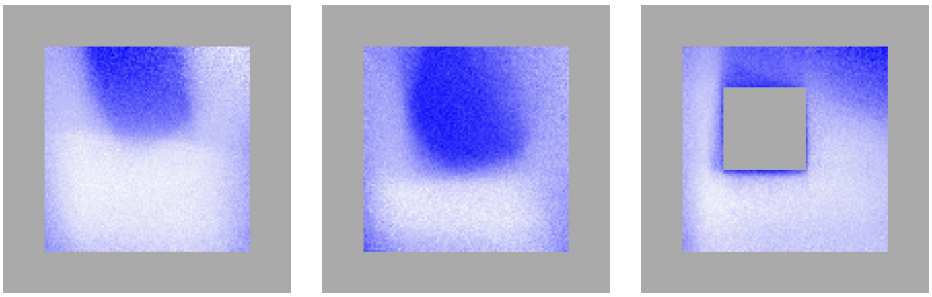
Table 3 and 4 shows the average return and policy explicability score of EPS compared to the baselines over 100 rollouts. The policy explicability score is computed using Equation (4.4) based on the synthetic human models. For EPS, the reconciliation parameter λ is tuned manually to show the different behaviors compared to the baselines in Figure 27, with $\lambda \in [2.0, 2.8]$. For all domains, EPS achieved the highest policy explicability score and followed the best performer for average return. PS performed well on most tasks while being considerably worse than others in D3. Due to the conflict between the human's expectation and task priorities in D3, the method failed to achieve a meaningful combination. DRLHP performed poorly on all



(a) Human (b) EPS (c) SAC (d) DRLHP (e) PS

Figure 27. Comparison of different learning methods with human's expectation (left). The dark grey area represents the walls. The brown area is the pit (with -100 penalty) and the green area is the goal (with +100 reward). The blue area represents icy roads and the yellow area represents sandy roads.

the domains because preference-based learning methods have difficulty dealing with domains with sparse or delayed rewards, which makes credit assignment challenging and results in large variations. Since EPS also applies a preference-based learning method to learn the surrogate reward function, we demonstrate this using the learned function. As shown in Figure 28, this function has a significant amount of uncertainty so is not ideal for guiding policy search. Although it adds variability into policy



(a) D1 (b) D2 (c) D3 & D4

Figure 28. Visualization of the learned surrogate reward functions for D1-D4. The darker the lower reward value and the brighter the higher.

Domain	EPS	SAC	DRLHP	PS
D1	95.53 (1.88)	**97.46 (1.53)**	74.51 (29.92)	93.91 (3.55)
D2	94.07 (2.43)	**95.65 (1.98)**	23.28 (43.71)	95.55 (3.17)
D3	**96.80 (1.16)**	94.47 (2.77)	-50.76 (56.49)	-178.76 (63.88)
D4	92.37 (1.80)	**93.27 (2.03)**	26.52 (49.02)	90.11 (14.79)

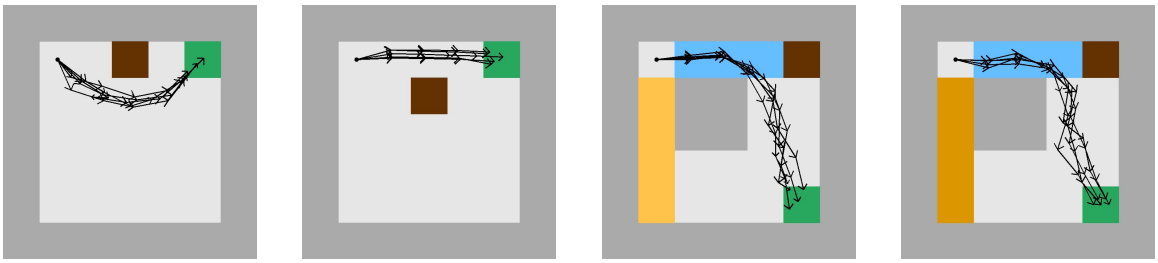
Table 3. Comparison of EPS to baselines using averaged return.

Domain	EPS	SAC	DRLHP	PS
D1	95.53 (1.88)	**97.46 (1.53)**	74.51 (29.92)	93.91 (3.55)
D2	94.07 (2.43)	**95.65 (1.98)**	23.28 (43.71)	95.55 (3.17)
D3	**96.80 (1.16)**	94.47 (2.77)	-50.76 (56.49)	-178.76 (63.88)
D4	92.37 (1.80)	**93.27 (2.03)**	26.52 (49.02)	90.11 (14.79)

Table 4. Comparison of EPS to baselines in terms of explicability score.

learning, it serves very well as an *auxiliary* objective by informing the agent where the human expects it to perform.

We show sampled trajectories for EPS, the baselines, and the human's expectation computed using the synthetic human models in Figure 27. In D1 and D2, SAC agent always chooses the shortest path while EPS agent takes a detour that bypasses the pit and hence is more explicable. In D3 and D4, SAC agent chooses the path passing



(a) Domain 1 (b) Domain 2 (c) Domain 3 (d) Domain 4

Figure 29. EPS agent's behaviors for different domains when there are no human biases.

through the top since the agent can get stuck on sandy roads in D3 and the top path is more efficient in D4. It is worth noting that EPS agent makes different decisions on these two domains. It selects the top path in D3 when the sandy road is difficult to navigate even though it is against the human's expectation. However, when it can better navigate on sand in D4, it chooses to respect the human's expectation to be more explicable, at the cost of lowering task performance. Policy shaping agent is also successful in D4, but gets stuck in D3, because simply multiplying two different policies could result in a new policy that is uninformative, irrespective of the λ used (e.g., when we have $p_1 = (0.1, 0.9), p_2 = (0.9, 0.1)$, the resulting policy from multiplying them would always be $p = (0.5, 0.5)$), which can lead to poor behaviors. In summary, we show that EPS can successfully generate effective behaviors that achieve an *intelligent* reconciliation between the human's expectation and task priorities. It can generate novel reconciled behaviors (D1 and D2), stick to the task priorities (D3), or follow the human's expectation (D4), as appropriately.

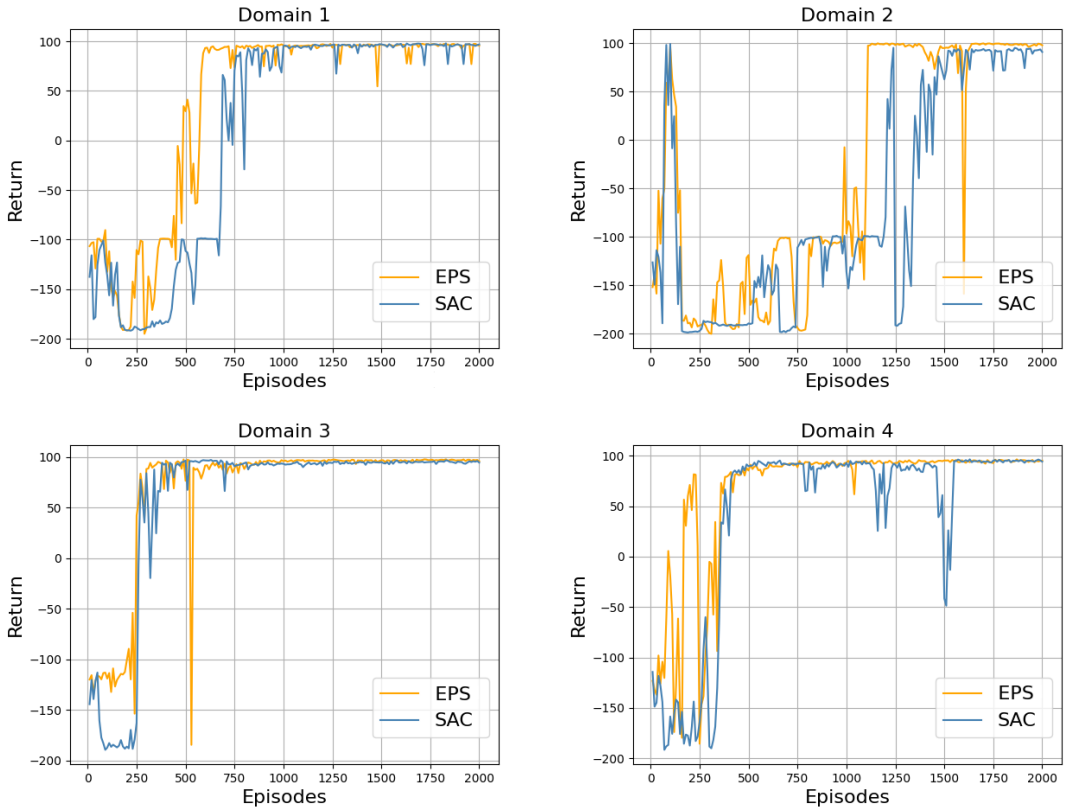


Figure 30. Comparison of the learning process of EPS and SAC in terms of return.

4.4.1.3 Synthetic Experiments Without Biases

We show that our method reduces to SAC when there is no human bias. The goal here is to examine that our method could recover the agent's optimal policy when human preference data is not biased. Figure 29 illustrates the behaviors of EPS agent without human biases. It successfully recovered the SAC agent's behavior as shown in Figure 27. We also present the cumulative rewards of EPS and SAC during the policy learning process in Figure 30. For all domains, we find that these two approaches are

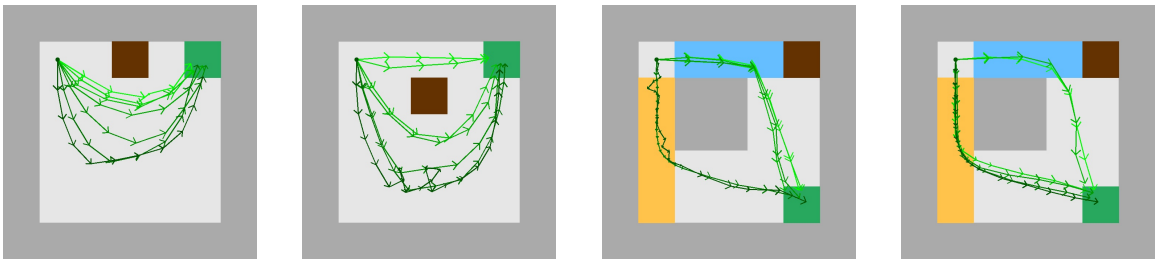
comparable (as expected) with EPS perhaps slightly better. EPS helped the agent learn faster since the surrogate reward function served now as a good inductive bias.

4.4.1.4 Sensitivity Analysis of the Reconciliation Hyperparameter

As shown in Equation (4.6), the hyperparameter λ reconciles between the reward from environment and the surrogate reward learned from the human data. To better understand the effect of this parameter algorithm, we consider a spectrum of the value of λ and compare the generated behaviors in all domains. We illustrate the generated behaviors with different λ in Figure 31. For all domains, it shows that the lower the λ, the closer the behavior is to the SAC agent, while the higher the λ, the closer the behavior is to the human's expectation. The range we tested is $\lambda \in [0.0, 3.0]$. Note that a similar weighting factor may arguably be used in policy shaping to achieve a similar reconciliation effect. However, policy shaping is fundamentally ill-posed (and hence much less robust) when the human's expected behavior and the optimal behavior differ significantly as seen in D3 from Figure 27. In real-world applications, this parameter should be set in a domain specific way, similar to the discussion in (Zhang et al. 2017; Kulkarni et al. 2019). In our work, we manually tune the λ value for each domain. The automatic tuning of it will be studied for future work.

4.4.2 Autonomous Driving Domain

We used a simulated autonomous driving domain (Leurent 2018) to evaluate our method with human subjects and demonstrate its real-world relevance, as illustrated in Figure 33. The state space is featured by the position and velocity of the ego-vehicle



(a) Domain 1 (b) Domain 2 (c) Domain 3 (d) Domain 4

Figure 31. EPS agent's behaviors with different λ in all domains. Light green trajectories have smaller λ values while dark green trajectories have larger λ values.

and nearby vehicles. The action space consists of five discrete actions. Initially, the autonomous driving vehicle (green) is running on the highway with a car in front of it (blue) on the same lane and with the same speed. The task of the autonomous driving agent is to handle situations when the front car slows down quickly and abruptly. One common type of cognitive bias is the *availability bias*, which reflects humans' tendency to overestimate the likelihood of events with greater availability in memory. Such biases can occur in driving since we regularly drive under familiar conditions. To design an experiment where such biases are present, we considered scenarios with a regular driving condition and a sleety condition where the vehicle's domain dynamics became more stochastic due to slippery roads. With the availability bias, however, humans are likely to make the same decisions under these similar but in actuality different conditions, leading to safety risks.

The user study consists of two phases: training and testing. At the beginning of the training phase, we requested the participates to provide the importance ratings for several factors governing the autonomous driving behavior (Cheung et al. 2018) in a 5-point Likert scale, such as average speed (3.42), distance to the front car (4.57), relative speed to neighboring cars (3.28), and lane following (3.78). The average participants' responses are shown in the parentheses above. Their responses served

as the engineered reward model (r_A) and are linearly combined. For example, the distance to the front car is directly associated with collision and hence given a larger weight. Before human data collection, we also presented the participants with the vehicle's behavior after braking under the regular condition when the front car slows down suddenly. Then, given the information that it is driving on a sleety day, we selected segments that showcased different vehicle behaviors after braking (i.e., braking behaviors under different effectiveness). The participants were asked to select which one matched their expectations the most. Their responses were used to validate the availability biases in this experiment. Then, we collected human data by actively selecting pairs of trajectory segments for the participants to compare with. The collected data was used to train our method and the baselines. Then, we showed rollouts of the learned policy for each method to new participants and asked them to rate those rollouts in the testing phase.

4.4.2.1 Results and Discussion

Our user study was published on the Amazon Mechanical Turk (MTurk) as shown in Figure 32 and 33. We recruited 15 participants for training with an average age of 39. Each of the participant was paid 1 dollar for this 20 minutes task. They were provided with instructions about the scenarios at the beginning. Next, they were asked to provide their importance ratings for several factors (features) regarding driving behaviors. After these questions, we asked the participants to select his or her preferred behavior from a pair of ego-vehicle's trajectory segments. The queries were

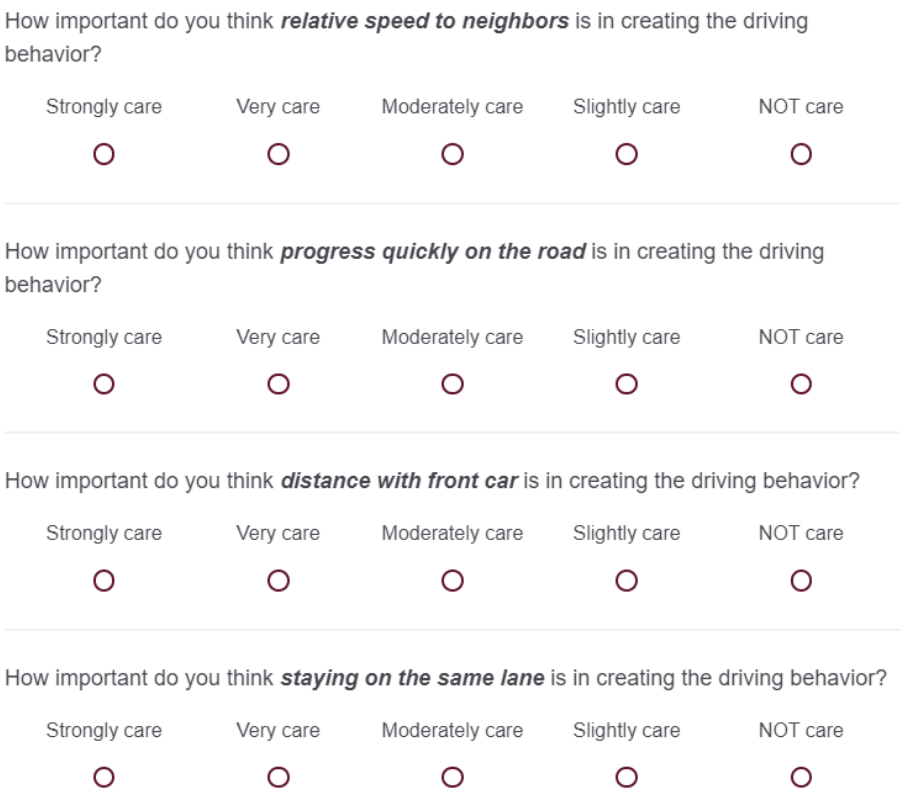


Figure 32. Solicit human's preference of several important feature regarding driving behaviors.

selected manually. The selected queries covered various scenarios, such as, slowing down by braking, immediately steering to the other lane, and so on.

To sift out invalid responses, a sanity check demonstration was added that showed a collision, which should not be preferred under any situation. We recruited 15 participants for training, and one failed the sanity check. For the bias validation question, 12 out of 14 valid participants chose that the ego-vehicle would slow down effectively on a sleety day when braking. This reflects the availability bias that the participants had, which can lead to collisions on a slippery road if ignored. We also noticed that the participants preferred to brake than steer in general during training,

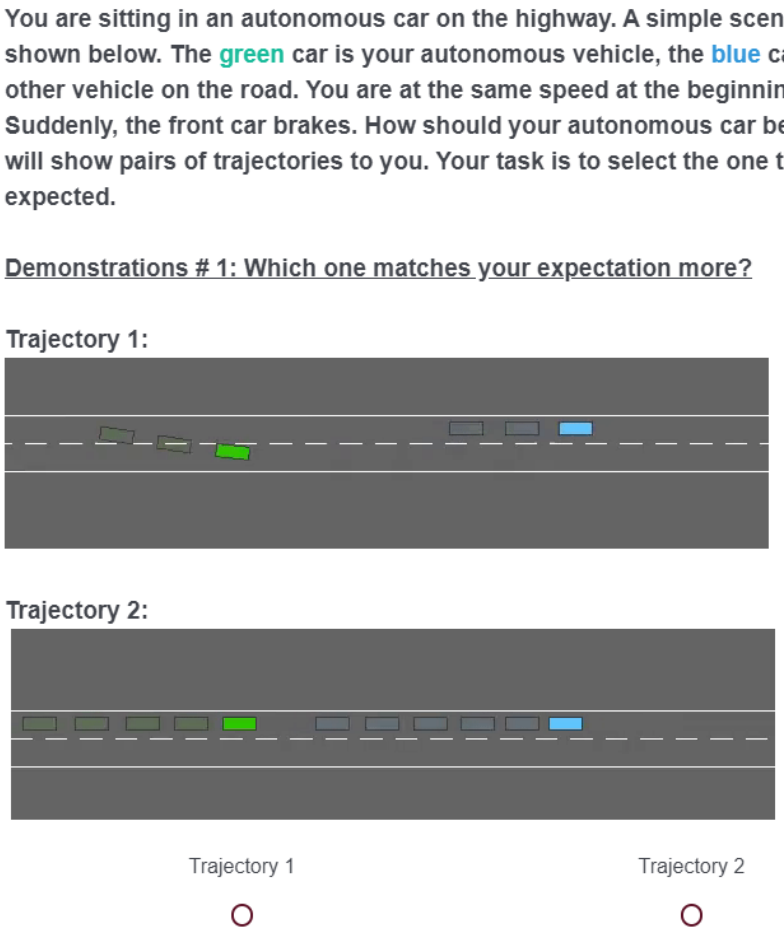


Figure 33. Scenario introduction and eliciting the human expectation-based preference.

	EPS	SAC	DRLHP
Avg. Rating	7.6	5.0	5.9
Avg. Return	11.0	11.9	9.6

Table 5. Comparison between EPS and baselines on an autonomous driving domain.

which accords with the participants' responses that attached a high importance rating to distance to the front car and lane following.

The testing phase occurs on a sleety day. The behavior of EPS, SAC, and DRLHP agents are illustrated in Figure 34, 35, and 36 respectively. SAC agent (see Figure 35)

steers immediately when the front car slows down. It is the most efficient behavior with the most return on a slippery road since braking would not be effective. DRLHP agent (see Figure 36), on the other hand, chooses to brake while staying in the same lane. Such a behavior, however, is more likely to lead to a collision (risk to human passengers). As we can see from Figure 36, the agent is getting dangerously close to the vehicle in the front. EPS agent (see Figure 34) first chooses to brake (to be explicable), and then steers to the other lane (to stay safe and continue moving forward), which maintains both explicability and task efficiency.

We demonstrated these rollouts in the testing phase to new participants. We informed the participants that the vehicle was running on a sleety day. Each participant was required to provide ratings to all the demonstrations ranging from 1 to 10. We obtained 15 valid responses. Interestingly, the participants rated EPS agent the most preferable, followed by DRLHP and SAC agents. The average rating for each agent and its standard deviation are shown in Figure 34, 35, and 36. We note that the results contradict with the importance ratings solicited for the engineered reward model, which should have led to preferring the steering behavior as chosen by the SAC agent on slippery roads. This further suggests that the participants had a biased belief about the vehicle's dynamics (i.e., braking is as efficient under the regular condition as under the sleety condition). Regardless, our agent chose the behavior that was both explicable and safe.

The reconciliation hyperparameter used to generate the explicable policy was 2.0. We presented results about the returns besides human ratings in Table 5. In terms of the reward function we obtained in the training phase, the SAC agent performs the best. In addition, DRLHP agent obtained a significantly lower return compared to the other two since it was more likely to collide with the front car when it chose

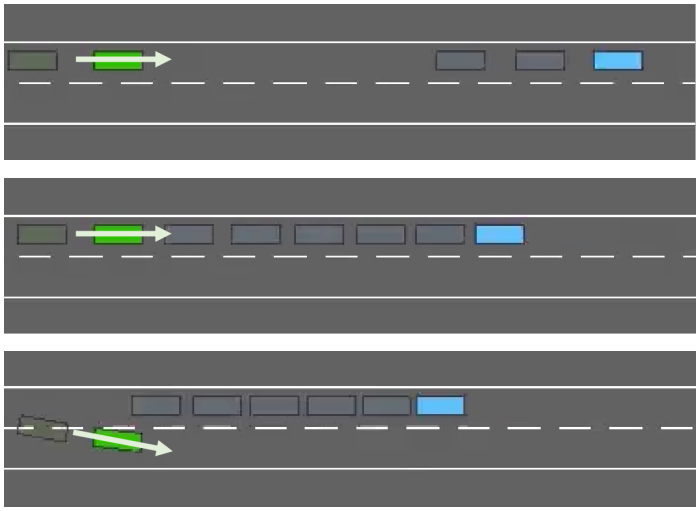


Figure 34. EPS agents' behaviors in the autonomous driving domain illustrated in three characteristic steps from the top to bottom for each agent. Its average rating and standard deviation are ($\mu = 7.6, \sigma = 2.4$).

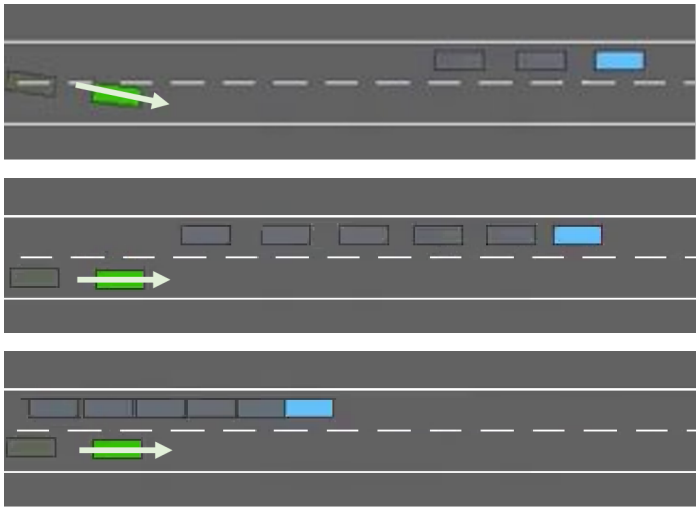


Figure 35. SAC agents' behaviors in the autonomous driving domain illustrated in three characteristic steps from the top to bottom for each agent. Its average rating and standard deviation are ($\mu = 5.0, \sigma = 2.3$).

to brake on a slippery road. Our EPS agent appropriately balanced in between the reward and human's expectation (that is biased).

Figure 36. DRLHP agents' behaviors in the autonomous driving domain illustrated in three characteristic steps from the top to bottom for each agent. Its average rating and standard deviation are ($\mu = 5.9, \sigma = 2.2$).

4.5 Conclusion

In this work, we introduced and formulated the problem of explicable policy search that considers model differences between the learning agent and its human observer. We developed an efficient solution by learning a surrogate reward function that was then used to recover an explicable policy. Our method significantly extends explicable planning to an RL setting and to stochastic environments with continuous state and action spaces. We evaluated in simulations and with human subjects. Results showed that our approach could better handle situations under model differences than several baselines and thus contributed a critical tool to achieving explainable human-agent interaction.

We assume that the human's (inaccurate) belief about the domain dynamics does not easily update as observations of the agent are made. For scenarios where inaccuracies stem from intrinsic cognitive biases that are difficult to change, this

is a reasonable assumption. In other cases where discrepancies were due to, e.g., information asymmetries, the belief can change dynamically and needs to be actively monitored (Hanni and Zhang 2021). Other possible directions include more complex combinations of the expected return and policy explicability score, as well as weighted policy explicability scores to incorporate trajectory similarity as considered in prior work (Zhang et al. 2017; Kulkarni et al. 2019).

Chapter 5

CONCLUSION

In this dissertation, we investigate the problem settings where the human's biased belief about the domain dynamics would influence the intelligent agent to learn the human's objectives, recover the optimal policy, and the way of behaving while working along with a human observer. Several approaches are present to address these problems that are raised by such model differences in human-aware decision making process.

In chapter 2, we introduce the Generalized Reward Learning (GRL) problem where the human biased belief about the domain dynamics may affect the reward learning, and propose a method called GeReL to recover the reward function while taking the human belief into consideration. To develop the method, we formulated the problem in a variational inference framework to learn the parameters governing the reward function and the human's belief about the domain dynamics simultaneously from observed human's ratings of agent demonstrations. We evaluated our approach experimentally using a simulated domain and with a user study whose results showed that GeReL outperformed prior approaches that could have misinterpreted the human objectives when such biases are not considered, and it could recover the true human objectives effectively even under such a challenging setting.

In chapter 3, we investigate the problem of preference-based reinforcement learning (PbRL) under human biased belief about the domain dynamics. Without consideration of the human belief, the learned surrogate reward function could be wrong and result in suboptimal or undesired behaviors. Similarly, we propose to learn a reward function while taking the human belief into account. To be applied on continuous control tasks,

both reward function and human belief are modeled using Bayesian neural networks. Their parameters are updated from the human preference data in a variational Bayesian framework. The learned reward function is then used to optimize the agent's policy. We evaluate and compare our method with a state-of-the-art PbRL baseline on several continuous control tasks. The results show that our method can successfully neutralize the belief biases and reach near-oracle performance compared to the baseline method.

In chapter 4, we introduced and formulated the problem of explicable policy search that considers model differences between the learning agent and its human observer. We developed an efficient solution that learns a surrogate reward function from human preferences between pairs of behavior segments. This learned surrogate reward function was then used to recover an explicable policy. Our method extends explicable planning to an RL setting and to stochastic environments with continuous state and action spaces. By evaluating in simulations and with human subjects, we show that our approach could better handle situations under model differences than several baselines and thus contributed a critical tool to achieving explainable human-agent interaction.

For future work, we believe that it is worth investigating how to learn an informative prior of the human biased belief in the work of reward learning and PbRL under human biased belief, especially for the model differences caused by common biases. We may learn a prior model through crowdsourcing. Such prior model could potentially captures the information of what the biases are like within a group of human users in certain scenarios and guide our approaches in chapter 2 and chapter 3 to recover the human belief and learn a reward function that describes the human's objectives much more efficiently and accurately. In addition, the learning frameworks proposed in chapter 2 and chapter 3 have to learn two models in the meanwhile. The complexity of

the learning process make it impractical for some real-world problems. A recent work on PbRL (Knox et al. 2022) introduces a new framework that learns a advantage function from the human's feedback which is leveraged directly in the policy optimization process. It contains information of the human's expectation. While it also doesn't consider the effects of the biased human belief on the learning process, the learned expectation would not be optimal and may be inapplicable for the learning agent to follow in the environment. It is still worth studying how to neutralize the belief biases without explicitly learning both a reward function and an estimated belief model. The learning of advantage function could be an interesting starting point. Also, consider that we have learned the human's expectation of the agent, whether we can borrow some insights from domain adaptation community to enable the agent adapt the learned expectation into the ground truth domain dynamics could be another promising direction of future work as well.

In addition, for all the work in this dissertation, we assume that the human's belief about the domain dynamics does not easily update as observations of the agent are made. For scenarios where the model differences stem from intrinsic cognitive biases that are difficult to change, this is a reasonable assumption. In other cases where discrepancies were due to, e.g., information asymmetries, the belief can change dynamically and needs to be actively monitored (Hanni and Zhang 2021). In turn, the agent's behavior could influence the human belief. In order to achieve a better teaming, the agent should be able to teach the human the correct domain dynamics by detecting the model differences and communicating the ground truth via its behaviors. Another possible direction of future work is regarding the reconciliation parameters in the framework of explicable policy search. We manually set it to be a fixed value, while the agent should be able to respect the human expectation of its behaviors

in various levels for different states and scenarios. Moreover, the model differences between the human belief and ground truth domain dynamics could be global (i.e., apply to the whole state space) or local (i.e., only for a subarea of the environment). To make the reconciliation factor dynamic in terms of how different are the models in each state is an attractive characteristic of the problem of explicable policy search and potentially a promising research direction that is worth investigating.

Furthermore, one of the limitations of our approaches proposed in this dissertation is the application of them to real-world problems and scenarios with real robots and humans. There exists several open challenges regarding human-agent interaction within the real-world tasks, such as, how frequent the intelligent agent should be interacting with the human users, how many human responses we need for learning, the needs for considering the impacts of fatigue or irrationality in human-agent interaction process, and how to effectively reduce the load from human, and so on. In addition, we need to consider the scenarios where the human and intelligent agents are working together (e.g., closely collaborative scenarios) which is pervasive for real-world tasks, rather than human as an observers only. We believe that research should not only be done in the simulations and discussed on papers. It should also be able to effectively applied to real-world tasks and benefit the real human users. Bridging the gaps between examining the proposed approaches in simulation and in real-world tasks with real robots and human users would also be an inspiring future goal we aim to achieve!

REFERENCES

Abbeel, Pieter, and Andrew Y Ng. 2004. "Apprenticeship learning via inverse reinforcement learning." In *Proceedings of the twenty-first international conference on Machine learning,* 1. ACM.

Akrour, Riad, Marc Schoenauer, and Michele Sebag. 2011. "Preference-based policy learning." In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases,* 12–27. Springer.

Armstrong, Stuart, and Sören Mindermann. 2018. "Occam's razor is insufficient to infer the preferences of irrational agents." In *Advances in Neural Information Processing Systems,* 5598–5609.

Baker, Chris, Rebecca Saxe, and Joshua Tenenbaum. 2011. "Bayesian theory of mind: Modeling joint belief-desire attribution." In *Proceedings of the annual meeting of the cognitive science society,* vol. 33. 33.

Bishop, Christopher M. 2006. *Pattern recognition and machine learning.* springer.

Blundell, Charles, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. 2015. "Weight uncertainty in neural network." In *International Conference on Machine Learning,* 1613–1622. PMLR.

Boularias, Abdeslam, Jens Kober, and Jan Peters. 2011. "Relative entropy inverse reinforcement learning." In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics,* 182–189.

Boutilier, Craig, Richard Dearden, and Moisés Goldszmidt. 2000. "Stochastic dynamic programming with factored representations." *Artificial intelligence* 121 (1-2): 49–107.

Bradley, Ralph Allan, and Milton E Terry. 1952. "Rank analysis of incomplete block designs: I. The method of paired comparisons." *Biometrika* 39 (3/4): 324–345.

Brockman, Greg, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. "Openai gym." *arXiv preprint arXiv:1606.01540.*

Busa-Fekete, Róbert, Balázs Szörényi, Paul Weng, Weiwei Cheng, and Eyke Hüllermeier. 2013. "Preference-based evolutionary direct policy search." In *ICRA Workshop on autonomous learning.*

Busa-Fekete, Róbert, Balázs Szörényi, Paul Weng, Weiwei Cheng, and Eyke Hüllermeier. 2014. "Preference-based reinforcement learning: evolutionary direct policy search using a preference-based racing algorithm." *Machine Learning* 97 (3): 327–351.

Chakraborti, Tathagata, Subbarao Kambhampati, Matthias Scheutz, and Yu Zhang. 2017. "Ai challenges in human-robot cognitive teaming." *arXiv preprint arXiv:1707.04775.*

Chakraborti, Tathagata, Anagha Kulkarni, Sarath Sreedharan, David E Smith, and Subbarao Kambhampati. 2019. "Explicability? legibility? predictability? transparency? privacy? security? the emerging landscape of interpretable agent behavior." In *Proceedings of the International Conference on Automated Planning and Scheduling,* 29:86–96.

Chakraborti, Tathagata, Sarath Sreedharan, Yu Zhang, and Subbarao Kambhampati. 2017. "Plan explanations as model reconciliation: moving beyond explanation as soliloquy." In *IJCAI,* 156–163. AAAI Press.

Chatterjee, Krishnendu, Rupak Majumdar, and Thomas A Henzinger. 2006. "Markov decision processes with multiple objectives." In *Annual Symposium on Theoretical Aspects of Computer Science,* 325–336. Springer.

Cheung, Ernest, Aniket Bera, Emily Kubin, Kurt Gray, and Dinesh Manocha. 2018. "Identifying driver behaviors using trajectory features for vehicle navigation." In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* 3445–3452. IEEE.

Christiano, Paul F, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. "Deep reinforcement learning from human preferences." In *Advances in neural information processing systems,* 4299–4307.

Clavera, Ignasi, Jonas Rothfuss, John Schulman, Yasuhiro Fujita, Tamim Asfour, and Pieter Abbeel. 2018. "Model-based reinforcement learning via meta-policy optimization." In *Conference on Robot Learning,* 617–629. PMLR.

Coumans, Erwin, and Yunfei Bai. 2021. *PyBullet, a Python module for physics simulation for games, robotics and machine learning.* http://pybullet.org.

Cui, Yuchen, and Scott Niekum. 2018. "Active reward learning from critiques." In *ICRA,* 6907–6914. IEEE.

Daniel, Christian, Malte Viering, Jan Metz, Oliver Kroemer, and Jan Peters. 2014. "Active Reward Learning." In *Robotics: Science and systems,* vol. 98.

Dorsa Sadigh, Anca D Dragan, Shankar Sastry, and Sanjit A Seshia. 2017. "Active preference-based learning of reward functions." In *RSS.*

Dragan, Anca D, Kenton CT Lee, and Siddhartha S Srinivasa. 2013. "Legibility and predictability of robot motion." In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI),* 301–308. IEEE.

Duchi, John, Elad Hazan, and Yoram Singer. 2011. "Adaptive subgradient methods for online learning and stochastic optimization." *JMLR* 12 (Jul): 2121–2159.

Erdem, B, Malayandi Palan, Nicholas C Landolfi, Dylan P Losey, Dorsa Sadigh, et al. 2020. "Asking easy questions: A user-friendly approach to active reward learning." In *Conference on Robot Learning,* 1177–1190. PMLR.

Eysenbach, Benjamin, Swapnil Asawa, Shreyas Chaudhari, Sergey Levine, and Ruslan Salakhutdinov. 2020. "Off-Dynamics Reinforcement Learning: Training for Transfer with Domain Classifiers." *arXiv preprint arXiv:2006.13916.*

Fox, Maria, and Derek Long. 2003. "PDDL2. 1: An extension to PDDL for expressing temporal planning domains." *Journal of artificial intelligence research* 20:61–124.

Gong, Ze, and Yu Zhang. 2018. "Behavior explanation as intention signaling in human-robot teaming." In *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN),* 1005–1011. IEEE.

———. 2020. "What is it you really want of me? generalized reward learning with biased beliefs about domain dynamics." In *Proceedings of the AAAI Conference on Artificial Intelligence,* 34:2485–2492. 03.

———. 2022a. "Explicable Policy Search." *Multidisciplinary Conference on Reinforcement Learning and Decision Making (RLDM).*

———. 2022b. "Neutralizing Belief Biases in Preference-based Reinforcement Learning." *In Submission.*

Griffith, Shane, Kaushik Subramanian, Jonathan Scholz, Charles L Isbell, and Andrea Thomaz. 2013. "Policy shaping: integrating human feedback with reinforcement learning." In *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2,* 2625–2633.

Gunning, David. 2017. "Explainable artificial intelligence (xai)." *Defense Advanced Research Projects Agency (DARPA), nd Web* 2 (2).

Gym Documentation. https://www.gymlibrary.ml/.

Haarnoja, Tuomas, Haoran Tang, Pieter Abbeel, and Sergey Levine. 2017. "Reinforcement learning with deep energy-based policies." In *Proceedings of the 34th International Conference on Machine Learning,* 1352–1361. JMLR.org.

Haarnoja, Tuomas, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor." *arXiv preprint arXiv:1801.01290.*

Haarnoja, Tuomas, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. 2018. "Soft actor-critic algorithms and applications." *arXiv preprint arXiv:1812.05905.*

Hanni, Akkamahadevi, and Yu Zhang. 2021. "Generating Active Explicable Plans in Human-Robot Teaming." In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* 2993–2998. IEEE.

Haring, Kerstin S, Katsumi Watanabe, Mari Velonaki, Chad C Tossell, and Victor Finomore. 2018. "FFAB—The Form Function Attribution Bias in Human–Robot Interaction." *IEEE Transactions on Cognitive and Developmental Systems* 10 (4): 843–851.

Hastings, W Keith. 1970. "Monte Carlo sampling methods using Markov chains and their applications."

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

Hinkley, David Victor, and DR Cox. 1979. *Theoretical statistics.* Chapman / Hall/CRC.

Hüllermeier, Eyke, Johannes Fürnkranz, Weiwei Cheng, and Klaus Brinker. 2008. "Label ranking by learning pairwise preferences." *Artificial Intelligence* 172 (16-17): 1897–1916.

Jordan, Michael I, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. 1999. "An introduction to variational methods for graphical models." *Machine learning* 37 (2): 183–233.

Kahneman, Daniel. 2011. *Thinking, fast and slow.* Macmillan.

Kingma, Diederik P, and Jimmy Ba. 2014. "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980.*

Kingma, Durk P, Tim Salimans, and Max Welling. 2015. "Variational dropout and the local reparameterization trick." *Advances in neural information processing systems* 28:2575–2583.

Knox, W Bradley, Stephane Hatgis-Kessell, Serena Booth, Scott Niekum, Peter Stone, and Alessandro Allievi. 2022. "Models of human preference for learning reward functions." *arXiv preprint arXiv:2206.02231.*

Knox, W Bradley, and Peter Stone. 2009. "Interactively shaping agents via human reinforcement: The TAMER framework." In *Proceedings of the fifth international conference on Knowledge capture,* 9–16.

Kulkarni, Anagha, Yantian Zha, Tathagata Chakraborti, Satya Gautam Vadlamudi, Yu Zhang, and Subbarao Kambhampati. 2019. "Explicable Planning as Minimizing Distance from Expected Behavior." In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems,* 2075–2077. International Foundation for Autonomous Agents and Multiagent Systems.

Kurutach, Thanard, Ignasi Clavera, Yan Duan, Aviv Tamar, and Pieter Abbeel. 2018. "Model-Ensemble Trust-Region Policy Optimization." In *International Conference on Learning Representations.*

Lee, Kimin, Laura Smith, and Pieter Abbeel. 2021. "PEBBLE: Feedback-Efficient Interactive Reinforcement Learning via Relabeling Experience and Unsupervised Pre-training." *arXiv preprint arXiv:2106.05091.*

Lee, Kimin, Laura Smith, Anca Dragan, and Pieter Abbeel. 2021. "B-Pref: Benchmarking Preference-Based Reinforcement Learning." *arXiv preprint arXiv:2111.03026.*

Leurent, Edouard. 2018. *An Environment for Autonomous Driving Decision-Making.* https://github.com/eleurent/highway-env.

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

Lillicrap, Timothy P, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. "Continuous control with deep reinforcement learning." *arXiv preprint arXiv:1509.02971.*

MacGlashan, James, Mark K Ho, Robert Loftin, Bei Peng, Guan Wang, David L Roberts, Matthew E Taylor, and Michael L Littman. 2017. "Interactive learning from policy-dependent human feedback." In *International Conference on Machine Learning,* 2285–2294. PMLR.

Ng, Andrew Y, and Stuart Russell. 2000. "Algorithms for Inverse Reinforcement Learning." In *in Proc. 17th International Conf. on Machine Learning.*

Oaksford, Mike, and Nick Chater. 2007. *Bayesian rationality: The probabilistic approach to human reasoning.* Oxford University Press.

Oudeyer, Pierre-Yves, Frdric Kaplan, and Verena V Hafner. 2007. "Intrinsic motivation systems for autonomous mental development." *IEEE transactions on evolutionary computation* 11 (2): 265–286.

Ramachandran, Deepak, and Eyal Amir. 2007. "Bayesian inverse reinforcement learning." In *IJCAI,* 2586–2591.

Ranganath, Rajesh, Sean Gerrish, and David Blei. 2014. "Black box variational inference." In *Artificial Intelligence and Statistics,* 814–822.

Reddy, Sid, Anca Dragan, and Sergey Levine. 2018. "Where Do You Think You're Going?: Inferring Beliefs about Dynamics from Behavior." In *Advances in Neural Information Processing Systems,* 1461–1472.

Robbins, Herbert, and Sutton Monro. 1951. "A stochastic approximation method." *The annals of mathematical statistics,* 400–407.

Roijers, Diederik M, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. 2013. "A survey of multi-objective sequential decision-making." *JAIR* 48:67–113.

Ross, S. M. 2002. *Simulation.* Elsevier.

Russell, Stuart J. 1998. "Learning agents for uncertain environments." In *COLT,* 98:101–103.

Sadigh, Dorsa, Anca D Dragan, Shankar Sastry, and Sanjit A Seshia. 2017. "Active Preference-Based Learning of Reward Functions." In *Robotics: Science and Systems.*

Schmidhuber, Jürgen. 2010. "Formal theory of creativity, fun, and intrinsic motivation (1990–2010)." *IEEE transactions on autonomous mental development* 2 (3): 230–247.

Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. "Proximal policy optimization algorithms." *arXiv preprint arXiv:1707.06347.*

Shah, Rohin, Noah Gundotra, Pieter Abbeel, and Anca Dragan. 2019. "On the feasibility of learning, rather than assuming, human biases for reward inference." In *International Conference on Machine Learning,* 5670–5679. PMLR.

Sigaud, Olivier, and Olivier Buffet. 2013. *Markov decision processes in artificial intelligence.* John Wiley & Sons.

Sutton, Richard S, and Andrew G Barto. 2018. *Reinforcement learning: An introduction.* MIT press.

Wang, Tingwu, Xuchan Bao, Ignasi Clavera, Jerrick Hoang, Yeming Wen, Eric Langlois, Shunshi Zhang, Guodong Zhang, Pieter Abbeel, and Jimmy Ba. 2019. "Benchmarking model-based reinforcement learning." *arXiv preprint arXiv:1907.02057.*

Wawrzyński, Paweł. 2009. "Real-time reinforcement learning by sequential actor–critics and experience replay." *Neural networks* 22 (10): 1484–1497.

Wawrzyński, Paweł, and Ajay Kumar Tanwani. 2013. "Autonomous reinforcement learning with experience replay." *Neural Networks* 41:156–167.

Wilson, Aaron, Alan Fern, and Prasad Tadepalli. 2012. "A bayesian approach for policy learning from trajectory preference queries." *Advances in neural information processing systems* 25:1133–1141.

Wirth, Christian, Riad Akrour, Gerhard Neumann, Johannes Fürnkranz, et al. 2017. "A survey of preference-based reinforcement learning methods." *JMLR.*

Wirth, Christian, and Johannes Fürnkranz. 2013a. "A policy iteration algorithm for learning from preference-based feedback." In *International Symposium on Intelligent Data Analysis,* 427–437. Springer.

———. 2013b. "EPMC: Every visit preference Monte Carlo for reinforcement learning." In *Asian Conference on Machine Learning,* 483–497. PMLR.

———. 2013c. "Preference-based reinforcement learning: A preliminary survey." In *Proceedings of the ECML/PKDD-13 Workshop on Reinforcement Learning from Generalized Feedback: Beyond Numeric Rewards.*

Zakershahrak, Mehrdad, Akshay Sonawane, Ze Gong, and Yu Zhang. 2018. "Interactive plan explicability in human-robot teaming." In *2018 27th IEEE International*

Symposium on Robot and Human Interactive Communication (RO-MAN), 1012–1017. IEEE.

Zhang, Jason, and Anca Dragan. 2019. "Learning from Extrapolated Corrections." In *ICRA,* 7034–7040. IEEE.

Zhang, Yu, Sarath Sreedharan, Anagha Kulkarni, Tathagata Chakraborti, Hankz Hankui Zhuo, and Subbarao Kambhampati. 2016. "Plan explicability for robot task planning." In *Proceedings of the RSS Workshop on Planning for Human-Robot Interaction: Shared Autonomy and Collaborative Robotics.*

———. 2017. "Plan explicability and predictability for robot task planning." In *ICRA,* 1313–1320. IEEE.

Ziebart, Brian D, J Andrew Bagnell, and Anind K Dey. 2010. "Modeling interaction via the principle of maximum causal entropy." In *ICML.*

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