LEARNING REPRESENTATIONS OF COGNITIVE DYNAMICS AND DECISION MAKING IN HUMAN DRIVERS

A Dissertation

by

RAN WEI

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Anthony McDonald
Co-Chair of Committee,	Alfredo Garcia
Committee Members,	Ceyhun Eksin
	Kenneth Easwaran
Head of Department,	Lewis Ntaimo

August 2023

Major Subject: Industrial Engineering

ABSTRACT

In order to function safely and autonomously, modern robotic systems need to understand other agents' mental states, including their beliefs and desires about the shared environment. This ability, known as Theory of Mind (TOM), is crucial for self-driving vehicles, as the exchange of beliefs through instantaneous maneuvers gives rise to nuanced social behavior in human driving and the lack of such exchange can lead to traffic conflicts and crashes. For social scientists and engineers, mental states serve as a compact representation of agent behavior, which can be used to understand human cognition, devise interventions on human cognitive limitations, and build autonomous agents to assimilate human behavior. However, the TOM ability in both humans and machines is not well-understood, with an important question being the unbounded possibility of agent beliefs leading to degenerate inference. In this dissertation, I study the possibility and advantages of TOM in the context of modeling human driving behavior. By proposing a set of algorithms to make inference about human drivers' mental states, I elicit the implicit assumptions in human's TOM ability. I show that human TOM likely involves a delicate balance between being realist about the environment and the unbounded imagination in a Bayesian fashion. These observations were engineered into the proposed algorithms, resulting in substantial improvements in interpreting abnormal human behavior, inspecting model failures, and robustifying control policies.

ACKNOWLEDGMENTS

This effort would not have been possible without the amazing professors I had during my undergraduate studies at Rutgers University. Most notably, professor MK. Jeong helped me find my interest in optimization and professor Missy Cummings (who was at Duke University at the time) helped me find my interest in understanding human behavior. Professor Weihong Guo introduced me to statistics and machine learning, for which I have carried my passion until this day and I am still in a constant pursuit of understanding. I still remember the day professor James Luxhoj convinced me to pick Industrial Engineering as my major and professor Mohsen Jafari and professor Susan Albin taught us the importance of public speaking and self-promotion. There are countless other faculties, staff, and TAs at Rutgers who have helped me; they wouldn't have known this day had come, but they know for sure they haven't seen the best of me yet.

I regret that I won't be able to properly acknowledge all my friends from before college, Rutgers, A&M, and other places, who have helped me develop as a person and provided me with constant support and challenges. However, this thesis could have been killed in the cradle if the following three folks did not show up. Hana Alambeigi is the first person I knew at A&M and has been helping me since day 1. Many things I do follow her footsteps. Yibo Zhu set my expectations right for the PhD program. His sharp perspectives keep me grounded with the reality. Rohith Karthikeyan jump-started my personal pursuit for ML and AI. I can now confidently say that I have listened to more ML podcasts than him. I am also fortunate to have Hannah and Josh Morales nearby. The annual visits to them remind me that PhD research is not everything.

I would also like to acknowledge my advisors Tony McDonald and Alfredo Garcia, and my committee members Ceyhun Eksin and Kenny Easwaran. I am extremely fortunate to have received the most freedom amongst the peers around me to explore the topics that interest me; at the same time, Tony and Dr. Garcia always seem to have endless availability for discussions. I want to thank Ceyhun and Kenny for their helpful feedback. I am always impressed by both their breadth of knowledge and depth of thinking.

It has also been a great pleasure working with Johan Engström and Matt O'Kelly among others from the Waymo safety team and Gustav Markkula from University of Leeds. They put on a show suggesting out-of-the-box ideas and they certainly do not hold back on their comments.

Most importantly, the source of this effort comes from my family. I have spoken most of the phone calls into existence. I believe this will also be true for everything that comes after.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Professors Anthony McDonald (advisor), Alfredo Garcia (co-advisor), Ceyhun Eskin of the Department of Industrial and Systems Engineering, and Kenneth Easwaran of the Department of Philosophy.

All work conducted for the dissertation was completed by the student independently under the advisorship of Professors Anthony McDonald and Alfredo Garcia.

Funding Sources

Graduate study was supported by a research assitantship from Texas A&M University.

This work was also supported in part by the following organizations:

- Department of Transportation, University Transportation Centers Program, Safety through Disruption University Transportation Center (451453-19C36)
- Army Research Office (W911NF1910201)

Any opinions, findings, conclusions, or recommendations expressed in this material are solely the responsibility of the author and do not necessarily reflect the official views of the funding organizations.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

With the increasing popularity and capability of machine learning (ML) models in consumer-facing and industrial applications, there is a growing demand for ML models and systems to be transparent, reliable, and aligned with human values. Despite already excelling at a variety of tasks, including image recognition [1, 2], machine translation [3], video game playing [4], among others, machine learning models are known to exhibit a handful of undesirable behavior, such as exploiting designed objectives and learning spurious correlations, resulting in biased decisions [5]. One way to improve current machine learning systems is to develop models that learn and behave like humans [6]. It is believed that humans learn and behave by building models of the world with intuitions of physics and psychology [7], giving rise to complex but generalizable and robust behavior.

One field with a particular interest in human-like models is automated and autonomous driving, not only because automated vehicles (AV) have to interact with human drivers, but also that human-like models and driving behavior are likely able to solve current challenges in AVs. Aside from the difficulty of building reliable perception systems, modern AVs are known to misbehave in traffic situations which require nuanced social behavior [8], such as behavior that communicate driver internal state (e.g., state of attention or aggressiveness) and intent (e.g., intent to overtake). The lack of such communicating behavior can lead to inefficient traffic in the best case, and in the worst case crashes as a result of other drivers' confusion [9].

The ability to understand other drivers' state and intent is known as theory of mind (TOM), an ability possessed by humans as early as 4 years old [10]. Computational models of TOM formulate such understanding as making inferences about agents who make rational decisions – decisions that realize their intent or desire – with respect to their beliefs about the world [11]. These models attempt to extract agent belief and desire from sensory data into compact representations (such as parameter vectors) for pattern analysis or predictions of future behavior, as well as learning signals for controlling similar autonomous systems.

Given that human beliefs and the process of forming them are potentially biased or misaligned with the actual environment, which leads to nuanced behavior that can be understood as naivete, illusion, or heuristics [11, 12, 13], belief inference is an important aspect of human TOM. However, most existing TOM frameworks solely focus on desire inference and assume humans always have accurate beliefs [14]. This dissertation aims to bridge this gap and understand the benefits of belief inference for both acquiring representations of agents and controlling autonomous systems.

1.1 Thesis Structure And Contributions

This dissertation studies belief inference in human TOM. By proposing and analyzing a set of TOM inference algorithms, I show the benefits of performing belief inference on two fronts:

- **Inference**: belief inference enables extracting novel insights of human driving behavior from observational data.
- **Control**: belief inference enables engineering transparent and robust control policies.

The dissertation document is organized as follow:

Part I: Background. This part provides relevant background for the thesis and identifies challenges faced by existing approaches to agent development.

- **Chapter 2** reviews models of rational decision-making under uncertainty with a focus on overcoming the interference between learning to perceive and learning to control in uncertain environments via joint perception-control modeling. I identify active inference as a promising joint modeling framework, provide a throughout review, and discuss challenges faced by existing active inference formulations.
- **Chapter 3** reviews Bayesian Theory of Mind as a framework for understanding agent behavior through their beliefs and desires. I provide an in-depth discussion on the identifiability of BTOM's and demonstrate the impossibility of recovering agent belief and desire under the naive BTOM formulation. I then suggest an approach to alleviate the un-identifiability by formulating informed priors.

Part II: Methods. In this part, I use three studies to illustrate the advantages of BTOM and active inference for inference and control.

- **Chapter 4** attempts to understand the advantages of active inference and BTOM in explaining human driving behavior. I propose a framework for inferring latent human beliefs and goals and interpreting the inferred parameters using dimensionality reduction techniques. Applying the framework to human emergency responses to automated vehicle failure reveals novel connections between trust and situation awareness and subjective beliefs. This chapter is reproduced from [15].

- **Chapter 5** attempts to understand the advantages of active inference and BTOM in terms of control performance and interpretability. I benchmark the active inference model against standard rule-based and black-box driver behavior models using a public highway car-following dataset. The results show that the active inference model not only outperforms rule-based and black-box models due to higher flexibility and more inductive biases, it also enables model introspection and editing due to its interpretable input-output mechanism. This chapter is reproduced from [16].

- **Chapter 6** investigates the performance advantage of control policies obtained by BTOM from data following the observations in Chapter 5. I analyze the objective function optimized by the BTOM algorithm and show that under a family of accuracy-promoting prior, BTOM corresponds to a robust inference problem. The family of priors are equivalent to the priors suggested in Chapter 2. Using these insights, I propose two scalable BTOM algorithms and show that they outperform state-of-the-art learning from demonstration algorithms without ad hoc engineering efforts. This chapter is reproduced from [17].

Part III: Conclusions. In this part, I conclude the thesis and discuss future directions.

2. ACTIVE INFERENCE: A JOINT MODEL OF PERCEPTION AND ACTION

2.1 Introduction

This chapter introduces active inference as a model of human perception and action. Active inference [18] is a recently proposed framework for modeling human behavior with growing popularity in neuroscience, philosophy, machine learning, economics, among other fields [19, 20, 21]. Derived from the Free Energy Principle (FEP) [22], active inference aims to provide a unified view of perception, action, and learning. Many prior models of human behavior have adopted a decoupled view, casting perception and action as separate and independent modules [23, 24]. In contrast, active inference proposes to understand the role of perception and action under a single imperative to minimize free energy, an information theoretic notion of surprise [22]. The benefit of this unification is a more nuanced understanding of the coupling roles of perception and action and potentially better explanation of human behavior.

Throughout this dissertation, I use the definition of perception as adjusting an agent's internal states in response to sensory signals via an embodied model, which is usually equated to making inference about the cause of signals, following Helmholtz's notion of unconscious inference [25]. I will use Partially Observable Markov Decision Process (POMDP) [26] as a minimal and yet expressive perception-action loop. POMDP has been extensively studied in the context of stochastic optimal control with well-established solution methods. A central problem in POMDP is the handling of epistemic uncertainty by trading off *exploration* and *exploitation*. Exploration refers to resolving uncertainty and learning about the unknown environment, while exploitation refers to committing to a course of rewarding actions at the cost of potentially greater loss due to ignorance. Active inference aims to provide an optimal handling of exploration and exploitation by decomposing its objective and the resulting behavior into two recognizable parts. This is in contrast to traditional POMDP solution methods, which implicitly handle the exploration-exploitation trade off [27]. While a few prior studies have tried to connect active inference with traditional POMDP

solution methods [28, 29] and understand how its exploratory behavior arise [30, 31] and is characteristic of active inference [32], the motivation and pragmatic benefit of the active inference formulation is far from clear [33].

Thus, the goal of this chapter is to synthesize the literature on active inference in POMDPs. I start by introducing POMDP and traditional approaches to behaving in either known or unknown POMDP environments. I then provide a concrete formulation of active inference based on existing literature and connect it to traditional POMDP approaches. I end with a discussion of the motivation for active inference and the open question of an unifying objective for autonomous agents.

2.2 Partially Observable Markov Decision Process

POMDP is a model of dynamic decision making under uncertainty that provides a minimally accurate depiction of a perception-action loop [26]. It assumes an agent has a model of the environment characterized by a set of states $s \in \mathcal{S}$, actions $a \in \mathcal{A}$, and observations $o \in \mathcal{O}$, all of which may be discrete or continuous. Upon receiving an action from the agent, the environment transitions to the next state according to probability distribution $P(s_{t+1}|s_t, a_t)$, where t indexes discrete time steps. The agent cannot directly perceive the environment state but register an observation signal emitted from the environment according to $P(o_t|s_t)$. This interactive process is called a controlled hidden Markov process (CHMP) with a schematic shown in Fig. 2.1. When the observation space is defined on the same set of symbols as the state space and provides precise information about the underlying state, i.e., $P(o_t|s_t) = \delta(o_t - s_t)$ where δ denotes the dirac delta function, such that there is no uncertainty about the state, this special instance of POMDP is called the Fully Observable Markov Decision Process, or Markov Decision Process (MDP).

POMDP usually assumes the agent has a true model of the environment and receives a reward $R(s, a)$ when taking action a in state s [34]. Agent behavior is governed by a policy $\pi(a)$, defined as a probability distribution over action $a \in \mathcal{A}$, with the goal of maximizing the sum of future

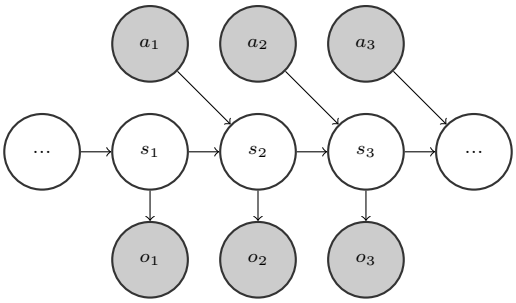


Figure 2.1: Bayesian network of a controlled hidden Markov process. Observable variables are colored in gray and hidden variables are transparent.

rewards for a planning horizon $H < \infty$, expected under future environment transitions:

$$\max_{\pi} \mathbb{E}_{P,\pi} \left[\sum_{\tau=t}^{t+H} R(s_\tau, a_\tau) \right] \tag{2.1}$$

where the τ denotes the time steps along the planning horizon rather than the agent's lifetime t in the environment. The solution to this problem is called the optimal policy denoted with π^* and the process through which the agent arrives at the optimal policy is called planning.

While the agent may choose from any class of policies, the optimal policy for a POMDP has to depend on the entire agent-environment interaction history $h_t = \{o_{1:t}, a_{1:t-1}\}$ in order to fully characterize the unknown state. Perception is used when the agent characterizes the history with its belief, defined as a probability distribution of the current environment state, i.e., $b(s) = P(s|h)$. The belief is a well-known sufficient statistic for characterizing history, making predictions, and generating optimal control [34]. This leads to a well-known class of exact solution method introduced in the next section.

2.2.1 Exact POMDP Solution Principle

The objective in (2.1) is not directly solvable since the environment state is unknown. A popular surrogate objective weights the reward by an estimate of the environment state: $R(h, a) \triangleq \sum_s P(s|h)R(s, a)$, where $P(s|h)$ is the posterior belief about the unknown state inferred from the CHMP model upon observing the history. Let us denote $ha_t = \{h_t, a_t\}$, the surrogate objective,

along with the expanded expectation, is written as:

$$\max_{\pi} \mathbb{E}_{\substack{P(o_{t+1:t+H}|ha_{t:t+H-1}) \\ \pi(a_{t:t+H}|h_{t:t+H})}} \left[\sum_{\tau=t}^{t+H} R(h_\tau, a_\tau) \right] \tag{2.2}$$

Denoting (2.2) with $V(h_t|\pi)$ and using the Markov property of the transition of histories, the objective can be decomposed as:

$$\begin{aligned} V(h_t|\pi) &= \mathbb{E}_{\substack{P(o_{t+1}|h_t,a_t) \\ \pi(a_t|h_t)}} \left[R(h_t, a_t) + \mathbb{E}_{\substack{P(o_{t+2:t+H}|ha_{t+1:t+H-1}) \\ \pi(a_{t+1:t+H}|h_{t+1:t+H})}} \left[\sum_{\tau=t+1}^{t+H} R(h_\tau, a_\tau) \right] \right] \\ &= \mathbb{E}_{\substack{P(o_{t+1}|h_t,a_t) \\ \pi(a_t|h_t)}} \left[R(h_t, a_t) + V(h_{t+1}|\pi) \right] \end{aligned} \tag{2.3}$$

This equation is referred to as the value or reward-to-go function under policy π [35]. The first term captures the immediate reward and the second term is the expected value in subsequent time steps following the same policy.

The optimal value function $V^*(h_t|\pi)$, alternatively written as $V(h_t)$, satisfies:

$$V(h_t) = \max_{\pi} \mathbb{E}_{\substack{P(o_{t+1}|h_t,a_t) \\ \pi(a_t|h_t)}} \left[R(h_t, a_t) + V(h_{t+1}) \right] \tag{2.4}$$

This decomposition, known as the Bellman optimality equation [35], suggests an efficient recursive computation of the optimal value function through a backward dynamic programming algorithm called value iteration, where the base case corresponds to the final time step of the planning horizon with expected value equals to the expected immediate reward.

A major challenge of solving a POMDP is representing the entire history of observations and actions required in the computation of (2.4). However, the entire history can be compactly represented by the posterior belief distribution, since it yields the same predictive distribution of the

next observation as explicitly storing the entire history:

$$\begin{aligned} P(o_{t+1}|h_t, a_t) &= \sum_{s_t} P(s_t|h_t) \sum_{s_{t+1}} P(s_{t+1}|s_t, a_t) P(o_{t+1}|s_{t+1}) \\ &\triangleq P(o_{t+1}|b_t, a_t) \end{aligned} \tag{2.5}$$

where $b_t \triangleq b(s_t|h_t) \triangleq P(s_t|h_t)$ denotes the posterior belief distribution, which can be computed recursively using the Bayes rule as:

$$b(s_{t+1}|h_{t+1} = \{h_t, o_{t+1}, a_t\}) = \frac{P(o_{t+1}|s_{t+1}) \sum_{s_t} P(s_{t+1}|s_t, a_t) b(s_t|h_t)}{\sum_{s_{t+1}} P(o_{t+1}|s_{t+1}) \sum_{s_t} P(s_{t+1}|s_t, a_t) b(s_t|h_t)} \tag{2.6}$$

Thus, the solution to the optimization problem in (2.2) can be found in the space of belief-action policies $\pi(a|b)$ and values instead of those conditioned on histories:

$$V(b_t) = \max_{\substack{\pi \\ \pi(a_t|b_t)}} \mathbb{E}_{P(o_{t+1}|b_t, a_t)} \left[R(b_t, a_t) + V(b_{t+1}) \right] \tag{2.7}$$

where b_{t+1} is the belief the agent would have updated to had it received observation o_{t+1} counterfactually given current belief b_t and action a_t.

It is often useful to define the (optimal) value function of taking an action a_t now and following the optimal policy π in subsequent time steps, because doing so gives the desirability of different actions immediately:

$$Q(b_t, a_t) = R(b_t, a_t) + \mathbb{E}_{P(o_{t+1}|b_t, a_t)} \left[V(b_{t+1}) \right] \tag{2.8}$$

This is known as the action-value or Q function [36]. This representation of the value function provides a simple method for finding the optimal policy and selecting actions:

$$\pi^*(a_t|b_t) = P\left(a_t = \arg\max_a Q(b_t, a) \right) \tag{2.9}$$

The formulation presented in this section is usually referred to as the belief MDP, since the

agent is planning with respect to the expected transitions in the belief space [26]. The benefit of the belief MDP is a compact representation of histories, which is easy to store and in principle admits dynamic programming solutions. However, in practice, solving the belief MDP is difficult because of the complexity of representing continuous belief states and belief-value functions, evaluating predictive distributions, and performing counterfactual belief updates.

2.2.2 Value Function Approximation

The purpose of value function approximation in POMDPs is to simplify its representation and computation to achieve a desired level of accuracy compared to the exact value function. This section reviews two types of common approximation methods: 1) a linear approximation which does not change the value function update method defined in (2.7), and 2) a heuristic approximation which further simplifies the defined update method.

2.2.2.1 Linear Value Function Approximation

In the discrete state setting, the value function can be approximated with a set of vectors $\Gamma = \{\alpha \in \mathbb{R}^{|\mathcal{S}|}\}$ such that the value function can be computed as [34]:

$$V(b) = \max_{\alpha \in \Gamma} \sum_{s \in \mathcal{S}} b(s)\alpha(s) \tag{2.10}$$

where $\alpha(s)$ denotes the s^{th} element of the α vector.

Using this approximation, the optimal value function can be computed by performing dynamic programming in the α vector space [37]:

$$V(b) = \max_{a \in \mathcal{A}} \left[\sum_{s \in \mathcal{S}} b(s)R(s,a) + \sum_{o' \in \mathcal{O}} P(o'|b,a) \max_{\alpha' \in \Gamma} \sum_{s' \in \mathcal{S}} b(s')\alpha'(s') \right] \tag{2.11}$$

with the α vectors updated as:

$$\alpha(s) = R(s,a^*) + \sum_{o' \in \mathcal{O}} P(o'|s,a^*) \max_{\alpha' \in \Gamma} \sum_{s' \in \mathcal{S}} b(s')\alpha'(s') \tag{2.12}$$

a^* refers to the action achieving the maximum in (2.11). The updated α vectors are added to the candidate set after each iteration.

While this method is faithful to the value function update rule defined in (2.7) and is considered the true value function in the ideal setting, it suffers from the difficulty of identifying the set of candidate α vectors and the growth of the candidate set [38, 39], which requires additional strategies to identify and prune [40, 41, 37].

2.2.2.2 Heuristic Value Function Approximation

A lingering challenge of the linear approximation method in (2.11) lies in evaluating the counterfactual observations and computing the subsequent beliefs, especially when the state and observation space is large. One way to overcome this challenge is to replace the counterfactual observations with counterfactual states [42]:

$$V(b) = \max_{a \in \mathcal{A}} \sum_{s \in \mathcal{S}} b(s) \left[R(s,a) + \sum_{s' \in \mathcal{S}} P(s'|s,a) \max_{\alpha' \in \Gamma} \alpha'(s') \right] \tag{2.13}$$

with the α vector updated as:

$$\alpha(s) = R(s,a^*) + \sum_{s' \in \mathcal{S}} P(s'|s,a^*) \max_{\alpha' \in \Gamma} \alpha'(s') \tag{2.14}$$

We can thus interpret the α vectors as the value function of a fully observable MDP with the same transition distribution and reward function as the belief MDP. We can also interpret the approximation as the assumption that the environment will become fully observable in the next time step.

This method is known as the QMDP approximation [42], because the approximate Q function of the belief MDP can be written in terms of the Q function of the underlying MDP denoted as $Q_{\text{MDP}}(s,a)$:

$$Q(b,a) = \sum_{s \in \mathcal{S}} b(s) Q_{\text{MDP}}(s,a) \tag{2.15}$$

Given the simplicity of computing the value function in the fully observable discrete state

setting, the QMDP method can be scaled to much larger state space than linear function approximation. However, it provides a less precise approximation of the true value function. As shown in [39], it is an upper bound of the linear approximation:

$$
\begin{aligned}
V_{\text{Linear}}(b) &= \max_{a\in\mathcal{A}} \left[\sum_{s\in\mathcal{S}} b(s)R(s,a) + \sum_{o'\in\mathcal{O}} P(o'|b,a) \max_{\alpha'\in\Gamma} \sum_{s'\in\mathcal{S}} b(s')\alpha'(s') \right] \\
&= \max_{a\in\mathcal{A}} \sum_{s\in\mathcal{S}} b(s) \left[R(s,a) + \sum_{o'\in\mathcal{O}} \max_{a'\in\Gamma} \sum_{s'\in\mathcal{S}} P(o'|s') \sum_{s\in\mathcal{S}} P(s'|s,a)b(s)\alpha'(s') \right] \\
&\leq \max_{a\in\mathcal{A}} \sum_{s\in\mathcal{S}} b(s) \left[R(s,a) + \max_{a'\in\Gamma} \sum_{s'\in\mathcal{S}} \sum_{s\in\mathcal{S}} P(s'|s,a)b(s)\alpha'(s') \right] \\
&\leq \max_{a\in\mathcal{A}} \sum_{s\in\mathcal{S}} b(s) \left[R(s,a) + \sum_{s'\in\mathcal{S}} P(s'|s,a) \max_{a'\in\Gamma} \alpha'(s') \right] \\
&= V_{\text{QMDP}}(b)
\end{aligned}
\tag{2.16}
$$

where the first inequality is due to removing the multiplier $P(o'|s')$ and the second inequality is due to $\max_\alpha \mathbb{E}[\alpha] \leq \mathbb{E}[\max_\alpha \alpha]$. In other words, the overestimation of the value function in QMDP is due to not accounting for the value reduction introduced by counterfactual observations. This is known as the value-of-information [43], or rather the cost of information.

2.3 Reinforcement Learning

So far we have considered cases where the agent has an accurate model of the environment which can be used for planning. However, this is often not the case for novice agents or agents in changing environments. In the more realistic setting, agents need to interact with the environment in order to build an adequate model of the environment, or directly search for a value maximizing policy. This interactive learning paradigm is called reinforcement learning (RL), with the explicit modeling-building approach called model-based reinforcement learning (MBRL) and the alternative approach called model-free reinforcement learning (MFRL).

2.3.1 Model-Free Reinforcement Learning

The goal of MFRL is to directly find the optimal policy without explicitly learning a model of the environment. The most straightforward method is to estimate the expected value after taking action a following a history h and take actions corresponding to the highest estimated value after seeing the same history the next time around.

The main concern with MFRL is to efficiently estimate the value function associated with a policy. Let $\mathcal{D} = \{(h_\tau, o_{\tau:\tau+H}, a_{\tau:\tau+H}, r_{\tau:\tau+H})\}_{\tau=1}^{t-H}$ denote a dataset of interaction history the agent has experienced while executing policy π in its lifetime and $Q_\theta(h_t, a_t|\pi)$ a function parameterizing the agent's estimate of the expected value associated with π with parameters θ, the agent may improve its estimate by minimizing the follow squared error:

$$\min_\theta \mathbb{E}_{h_t,a_t\sim\mathcal{D}} \left(Q_\theta(h_t, a_t|\pi) - \hat{Q}(h_t, a_t|\pi) \right)^2 \tag{2.17}$$

where $\hat{Q}$ is an empirical estimate of the value function associated with π constructed from samples in the dataset according to:

$$\hat{Q}(h_t, a_t|\pi) = \mathbb{E}_{(o_{t:t+H}, a_{t:t+H})\sim\mathcal{D}} \left[\sum_{\tau=t}^{t+H} R(h_\tau, a_\tau) \right] \tag{2.18}$$

However, a much more efficient way to estimate the expected value is to make use of one's existing estimates:

$$\hat{Q}(h_t, a_t|\pi) = R(h_t, a_t) + \mathbb{E}_{(o_{t+1}, a_{t+1})\sim\mathcal{D}} \left[Q_\theta(h_{t+1}, a_{t+1}) \right] \tag{2.19}$$

Alternatively, one may directly estimate the value of the optimal policy $\hat{Q}(h_t, a_t)$ leveraging the Bellman optimality equation:

$$\hat{Q}(h_t, a_t) = R(h_t, a_t) + \max_{a\in\mathcal{A}} \mathbb{E}_{o_{t+1}\sim\mathcal{D}} \left[Q_\theta(h_{t+1}, a) \right] \tag{2.20}$$

When estimates of (2.19) or (2.20) are used, the squared error in (2.17) is called the Bellman residual. Bellman residual minimization underlies the majority of state-of-the-art reinforcement learning methods, including the well-known Go-playing agent Alpha-Go [44]. It can be viewed as a type of approximate planning where the expected values and thus the associated policy are estimated by samples from a given environment.

2.3.2 Model-Based Reinforcement Learning

In MBRL, the agent builds an explicit model of the environment to facilitate planning. Given that a model can be introspected and simulated for alternative purposes, such as planning for a different set of rewards, model building is believed to be a central mechanism for human learning and adaptive behavior [7, 45, 6].

In the POMDP setting, the agent typically builds a model $P_\phi(o'|h,a)$ with parameters ϕ to predict the next observation o' based on the interaction history h and action a. Given past interaction experience contained in dataset $\mathcal{D}$, the agent can estimate, or learn, the model parameters by maximizing the log likelihood of the all observations in the dataset:

$$\max_{\phi} \sum_{\tau=1}^{t} \log P_\phi(o_{\tau+1}|h_\tau, a_\tau) \tag{2.21}$$

Instead of planning using samples drawn from past experience as in MFRL, model-based agents plan using samples drawn from their own model. The value estimates for model-based agents can be written as:

$$\hat{Q}(h_t, a_t) = R(h_t, a_t) + \mathbb{E}_{o_{t+1}\sim P_\phi(\cdot|h_t,a_t)}\left[\max_{a\in\mathcal{A}} Q_\theta(h_{t+1}, a)\right] \tag{2.22}$$

Thus, during the agent's lifetime, it interleaves collecting experience in the environment, updating the model estimate θ, and planning for the optimal policy. This process is described in Algorithm 1.

Algorithm 1 Model-based reinforcement learning

Require: Environment, model $P_\phi(o'|h,a)$, value estimate $Q_\theta(h,a)$, policy $\pi_\theta(h,a)$
  while $t \leq T$ **do**
    Interact with the environment
    Estimate model using (2.21)
    Estimate value using (2.22)
    Obtain optimal policy $\pi_\theta(a|h)$ from value estimates $Q_\theta(h,a)$
  end while

2.3.3 Objective Mismatch in MBRL

A central problem in MBRL is the objective mismatch between model estimation and planning in the sense that estimating a model using (2.21) does not directly contribute to planning better policies [46]. Often, such a mismatch leads to sub-optimal convergence due to a sub-optimal policy planned using an inaccurate model of the environment [47]. As Kearns & Singh showed in the well-known simulation lemma [48], the expected value a MBRL agent can obtain is bounded by the difference between its estimated model and the true model of the environment.

Although in principle, the difference between the estimated and the true model approaches zero with an expressive model and infinite data, leading to the ideal case of achieving the optimal value implied by the simulation lemma, in reality, this is infeasible. Disregarding the infinite lifetime requirement of infinite data, in practice, all models currently available for real-world MBRL do not satisfy the expressivity requirement and suffer from compounding error in long-horizon predictions [47, 46, 49]. It is also well established that humans do not build exact models of the environment – instead, human planning is largely based on intuition [6]. To this end, there is an increasing interest in the RL community to develop unifying objectives for joint model learning and policy planning beyond independent objective or optimization of model and policy [50, 51, 52, 53].

2.3.4 Reinforcement Learning as Probabilistic Inference

A well-known dilemma in RL is the trade off between exploiting current knowledge for maximizing reward and accumulating more knowledge about the environment via exploration. It usually makes sense to explore early in an agent's lifetime and switch to the exploit mode once sufficient

knowledge has been gathered. Thus, it is useful for the agent to know how to handle *epistemic uncertainty* – the uncertainty due to not knowing enough about the environment – either in the hidden state or unknown parameters.

A principled approach for handling epistemic uncertainty is via Bayesian inference. Given the objective mismatch in MBRL (Section 2.3.3), it is of interest whether Bayesian inference can benefit both model estimation and planning [54, 55]. The main challenge is formulating the joint model estimation-planning process as a probabilistic model with both observed and unobserved quantities such that the unobserved can be inferred using the Bayes rule. This section reviews RL as inference in both model-free and model-based settings.

2.3.4.1 Planning as Probabilistic Inference

Planning or control-as-inference [56, 57, 58, 59] is a planning method which reverses the usual question of what is the policy I should adopt in order to optimize expected value into *given that I behave optimally, what might be the policy I have taken*? It defines the optimality when selecting action a following a history h using a variable $\mathcal{O}$ with conditional probability:

$$P(\mathcal{O}|h,a) = \exp R(h,a) \tag{2.23}$$

It further assumes the agent has a model of the environment $P(o'|h,a)$ and an a priori policy $\pi(a)$. The joint distribution of a sequence of observations, actions, and the associated optimality variables for a horizon H can be written as:

$$\begin{aligned} &P(o_{t:t+H}, a_{t:t+H}, \mathcal{O}_{t:t+H}|h_{t-1}) \\ &= \prod_{\tau=t}^{t+H} P(\mathcal{O}|h_\tau, a_\tau) P(o_\tau|h_{\tau-1}, a_{\tau-1}) \pi(a_\tau) \\ &= \prod_{\tau=t}^{t+H} \exp R(h_\tau, a_\tau) P(o_\tau|h_{\tau-1}, a_{\tau-1}) \pi(a_\tau) \end{aligned} \tag{2.24}$$

with a Bayesian network illustration shown in Fig. 2.2.

By treating the optimality variables as observed and future observations and actions as hidden,

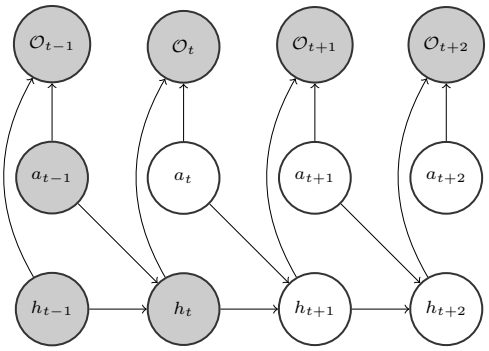


Figure 2.2: Planning-as-inference Bayesian network. Observed variables are colored in gray. Future optimality variables $\mathcal{O}$ are assumed to be observed while future histories and actions are assumed to be unobserved.

planning-as-inference corresponds to finding a posterior policy:

$$\pi(a_t|h_t, \mathcal{O}_{t:t+H}) = \frac{P(\mathcal{O}_{t:t+H}|h_t, a_t)\pi(a_t)}{P(\mathcal{O}_{t:t+H}|h_t)} \tag{2.25}$$

Using an approximate Bayesian inference technique (i.e., variational inference) to be introduced in Section 2.4.1, it can be shown that the posterior policy has the form:

$$\pi(a_t|h_t, \mathcal{O}_{t:t+H}) = \frac{\exp Q(h_t, a_t)}{\sum_{\tilde{a}} \exp Q(h_t, \tilde{a})} \tag{2.26}$$

where

$$\begin{aligned} Q(h_t, a_t) &= \max_{\tilde{\pi}} \mathbb{E}_{P,\tilde{\pi}} \left[\sum_{\tau=t}^{t+H} R(h_\tau, a_\tau) + \log \pi(a_\tau) - \log \tilde{\pi}(a_\tau|h_\tau) \middle| h_t, a_t \right] \\ &= R(h_t, a_t) + \log \pi(a_t) + \mathbb{E}_{P(o_{t+1}|h_t,a_t)} \left[V(h_{t+1}) \right] \\ V(h_t) &= \log \sum_{\tilde{a}} \exp Q(h_t, \tilde{a}) \end{aligned} \tag{2.27}$$

While resembling the value functions defined in Section 2.2.1, the Q, V, and π defined here have three major differences. First, instead of maximizing purely the expected cumulative rewards, the agent also maximizes the expected log likelihood of taking actions under the prior policy and

minimizes the expected log likelihood (negative entropy) of taking actions under the current policy. Thus, this method is also called maximum-entropy RL [60]. Second, instead of taking the maximum of the Q function, the value function defined here takes a *soft* maximum using the log-sum-exp function. Lastly, the policy randomizes over actions with probability proportional to the exponential of the Q values. These modifications encourage the agent to act as randomly as possible while conforming to decisions with maximum expected value and prior probabilities. In this way, the agent is less prone to sub-optimal convergence from committing to a single course of actions.

2.3.4.2 Joint Model Learning and Planning as Probabilistic Inference

The joint model learning and planning as inference framework falls under the Bayesian reinforcement learning paradigm, which treats both model learning and planning as Bayesian inference [61]. Instead of inferring a point estimate of the environment parameters θ as done in (2.21), Bayesian RL infers a full posterior distribution over the environment parameters from the dataset at every time step using the Bayes rule:

$$P(\theta|\mathcal{D}) = \frac{P(\mathcal{D}|\theta)P(\theta)}{P(\mathcal{D})} \tag{2.28}$$

Given the posterior distribution, the agent can predict and simulate future observations using the posterior predictive distribution:

$$P(o'|h,a) = \int_{\theta} P(o'|h,a;\theta)P(\theta|\mathcal{D}) \tag{2.29}$$

where the average (integral) over the posterior distribution better calibrates prediction uncertainty.

Planning in Bayesian RL is done by including the unknown parameters as a part of the hidden state space such that the hidden state dynamics factorize as:

$$P(o',\theta'|h,a;\theta) = P(o'|h,a;\theta')\delta(\theta'-\theta) \tag{2.30}$$

where the hidden parameters do not change over time. The agent can treat the new hidden dynamics as a special POMDP and find a policy using a method introduced in Section 2.2.1 and 2.2.2. Including unknown parameters as a part of the POMDP is known as Bayes-Adaptive POMDP [62, 63], with the corresponding Bayesian network shown in Fig. 2.3.

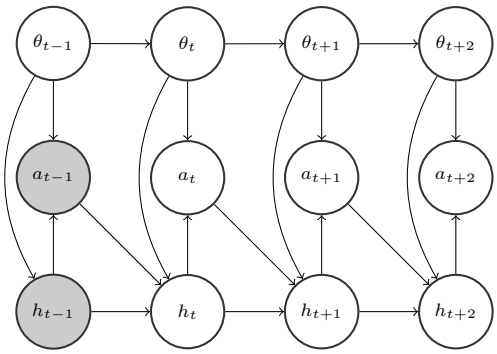


Figure 2.3: Bayes-Adaptive POMDP Bayesian network. Observed variables are colored in gray and unobserved variables are transparent. Transition of unknown parameters is $P(\theta'|\theta) = \delta(\theta' - \theta)$.

Although performing exact Bayesian inference and POMDP planning is extremely difficult, approximate Bayesian inference and planning has led to significant improvements in practice. In [49], the authors showed that representing uncertainty over the environment parameters using a Bayesian neural network can alleviate sub-optimal convergence. In [64], the authors showed that approximate Bayes-Adaptive POMDP agents exhibit highly intelligent exploration strategies.

2.4 Active Inference

Active inference is a framework for modeling perception and action derived from the Free Energy Principle [18]. The FEP states that behavior of living systems can be understood as minimizing surprisal or maximizing model evidence, defined as the negative log marginal likelihood of the signals they observe, $-\log P(o)$, under an embodied model of the environment [22]. Given this premise, the role of perception is to make inference about the hidden causes of observed signals, the role of learning is to build models of the hidden causal structure and parameters in the environment, and the role of action is to intervene on the environment such that future signals

are better predicted by the model. In this sense, active inference is a class of model-based reinforcement learning method. In contrast to the MBRL approach presented earlier, active inference makes a specific assumption that all parts of the perception-action process optimize a notion of free energy designed to overcome the intractability of calculating the log marginal likelihood. The attempt to unify perception and action gives active inference a special epistemic status compared to the traditional notion of MBRL.

The goal of this section is to review active inference. I start by introducing variational inference, the core inference and optimization method used by active inference, and then introduce the standard formulation of active inference based on [18] and [65]. I then connect active inference to traditional POMDPs and present a novel justification for the value-encoding choice made in active inference and its implications for scaling active inference. I end with a review of the neuroscience motivations for active inference and discuss potential advantages and concerns.

2.4.1 Variational Inference

Variational inference is a method proposed to overcome the intractability of performing exact Bayesian inference [66]. Consider performing posterior inference in a minimal probabilistic model $P(o) = \sum_s P(o|s)P(s)$ with latent variable s. Upon observing a sensory signal o, the exact inference over hidden cause s is given by the Bayes rule:

$$P(s|o) = \frac{P(o|s)P(s)}{\sum_s P(o|s)P(s)} \tag{2.31}$$

However, when the hypothesis space of hidden causes s is large, computing the sum in the denominator becomes intractable.

Variational inference is an alternative to the direct computation of the Bayes rule via the optimization of an approximate posterior distribution $Q(s)$ which minimizes the Kullback-Leibler

(KL) divergence to the true posterior distribution $P(s|o)$.

$$\begin{aligned}
&\min_{Q(s)} D_{KL}\left[Q(s)||P(s|o)\right] \\
&\quad = \mathbb{E}_{Q(s)}[\log Q(s) - \log P(s|o)] \\
&\quad = \mathbb{E}_{Q(s)}[\log Q(s) - \log P(s,o) + \log P(o)] \\
&\quad = \mathbb{E}_{Q(s)}[\log Q(s) - \log P(s,o)] + \log P(o)
\end{aligned} \tag{2.32}$$

The last line shows that the optimization can be performed without computing the intractable $\log P(o)$ since it does not depend on $Q(s)$. The remaining terms are called variational free energy (VFE), or free energy, denoted with $\mathcal{F}$:

$$\begin{aligned}
\mathcal{F}(o, Q) &= \mathbb{E}_{Q(s)}[\log Q(s) - \log P(s,o)] \\
&= -\log P(o) + D_{KL}\left[Q(s)||P(s|o)\right] \geq -\log P(o)
\end{aligned} \tag{2.33}$$

where the second line shows that it is an upper bound of the negative log marginal likelihood $-\log P(o)$. Thus, in statistics, free energy is also referred to as the (negative) evidence lower bound (ELBO). When $Q(s)$ is parameterized by a distribution class which contains $P(s|o)$, upon optimizing (2.32) to 0, free energy equals the negative log marginal likelihood.

2.4.2 The Perception-Action Loop in Active Inference

The standard active inference formulations in the literature consider agents with a finite lifetime T [18, 65]. The agent represents the complete hidden state and action sequence $s_{1:T}, a_{1:T-1}$ up to the final time and observations up to the current time t using a model with the following factorization [65]:

$$P(o_{1:t}, s_{1:T}, \pi) = \prod_{\tau=1}^{t} P(o_\tau|s_\tau) \prod_{\tau'=1}^{T} P(s_{\tau'}|s_{\tau'-1}, \pi) P(\pi) \tag{2.34}$$

where $\pi = a_{1:T-1}$ denotes the complete action sequence and $P(s_0|s_{0-1}, \pi) = P(s_0)$.

Using $Q(s_{1:T}, \pi) = Q(s_{1:T}|\pi)Q(\pi)$, where $Q(s_{1:T}|\pi)$ corresponds to the agent's estimate of

past and future states and $Q(\pi)$ corresponds to the policy to be enacted, the free energy function can be written as:

$$\begin{aligned}\mathcal{F}(o_{1:t}, Q) &= \mathbb{E}_Q \left[\log Q(s_{1:T}, \pi) - \log P(o_{1:t}, s_{1:T}, \pi)\right] \\ &= D_{KL}\left[Q(\pi)||P(\pi)\right] + \mathbb{E}_Q[\log Q(s_{1:T}|\pi) - \log P(o_{1:t}, s_{1:T}|\pi)] \\ &\triangleq D_{KL}\left[Q(\pi)||P(\pi)\right] + \mathbb{E}_{Q(\pi)}\left[\mathcal{F}(o_{1:t}, Q|\pi)\right]\end{aligned} \tag{2.35}$$

where $\mathcal{F}(o_{1:t}, Q|\pi)$ denotes action conditioned free energy. At every time step t, the agent first obtains an updated $Q(s_{1:T}|\pi)$ by minimizing (2.35) while fixing $Q(\pi)$ from the previous time step, and then updates $Q(\pi)$ by minimizing the same function while fixing $Q(s_{1:T}|\pi)$.

2.4.2.1 Perception in Active Inference

To illustrate how free energy minimization affects the agent's state estimates, we use a simple factorization of $Q(s_{1:T}|\pi) = \prod_{t=1}^{T} Q(s_t|\pi)$, known as the mean-field factorization [66]. This makes the action conditioned free energy equal to:

$$\mathcal{F}(o_{1:t}, Q|\pi) = \mathbb{E}_Q \left[\sum_{\tau=1}^{T} \log Q(s_\tau|\pi) - \sum_{\tau=1}^{t} \log P(o_\tau|s_\tau) - \sum_{\tau=1}^{T} \log P(s_\tau|s_{\tau-1}, \pi)\right] \tag{2.36}$$

Taking the derivative of $\mathcal{F}(o_{1:t}, Q|\pi)$ with respect to each $Q(s_\tau|\pi)$ and set to zero, we can show that the optimal state estimates satisfy (see Appendix A.1 for derivation):

$$\begin{aligned}\log Q^*(s_\tau|\pi) \propto \mathbb{I}[\tau \leq t] \log P(o_\tau|s_\tau) + \mathbb{E}_{Q^*(s_{\tau-1}|\pi)}[\log P(s_\tau|s_{\tau-1}, \pi)] \\ + \mathbb{E}_{Q^*(s_{\tau+1}|\pi)}[\log P(s_{\tau+1}|s_\tau, \pi)]\end{aligned} \tag{2.37}$$

where $\mathbb{I}[\cdot]$ is the indicator function.

This equation shows that the optimal state estimates have the form:

$$Q^*(s_\tau|\pi) \propto \begin{cases} \exp\left(\mathbb{E}_{Q^*(s_{\tau-1}|\pi)}[\log P(o_\tau, s_\tau|s_{\tau-1}, \pi)] + c\right), \tau \leq t \\ \exp\left(\mathbb{E}_{Q^*(s_{\tau-1}|\pi)}[\log P(s_\tau|s_{\tau-1}, \pi)] + c\right), \tau > t \end{cases} \tag{2.38}$$

where c is a term accounting for future states. In other words, optimal estimates of past states approximate exact Bayesian posterior distributions and optimal estimates of future states approximate exact Bayesian posterior predictive distributions.

2.4.2.2 Action in Active Inference

Given that state estimation via free energy minimization merely approximates exact Bayesian inference, active inference needs to encode value in its model in order to generate purposeful behavior. It does so by equipping the agent with a special prior over action sequences [67]:

$$P(\pi) \propto \exp(-\mathcal{G}(\pi|Q^*)) \tag{2.39}$$

where $\mathcal{G}(Q, \pi)$ is called the expected free energy (EFE) defined as [67]:

$$\mathcal{G}(\pi|Q^*) \triangleq \mathbb{E}_{Q^*(o_{t+1:T}, s_{t+1:T}|\pi)} \Big[\log Q^*(s_{t+1:T}|\pi) - \log \tilde{P}(o_{t+1:T}, s_{t+1:T}|\pi)\Big] \tag{2.40}$$

where $Q^*(o_{t+1:T}, s_{t+1:T}|\pi) = P(o_{t+1:T}|s_{t+1:T})Q^*(s_{t+1:T}|\pi)$ is the joint predictive distribution. The term $\tilde{P}$ defines a desired distribution over the hidden states and observations where the dependence on action sequences is usually ignored. For example, for a reward-driven agent with reward function $R(s)$, $\tilde{P}$ can be defined as [29]:

$$\tilde{P}(o_{t+1:T}, s_{t+1:T}|\pi) = \prod_{\tau=t+1}^{T} P(o_\tau|s_\tau) \frac{\exp R(s_\tau)}{\sum_{s'_\tau} \exp R(s'_\tau)} \tag{2.41}$$

In this way, we can interpret $\mathcal{G}(\pi|Q^*)$ as carrying the expected cumulative reward.

Given the definition of the EFE prior, the optimal action posterior minimizing $\mathcal{F}(o_{1:t}, Q)$ is:

$$Q^*(\pi) \propto \exp\left(-\mathcal{G}(\pi|Q^*) - \mathcal{F}(o_{1:t}, Q^*|\pi)\right) \tag{2.42}$$

2.4.2.3 *Properties of the Expected Free Energy*

Under the mean-field factorization and desired distribution $\tilde{P}$ defined according to (2.41), the EFE can be written as [65]:

$$
\begin{aligned}
\mathcal{G}(\pi|Q^*) &= \sum_{\tau=t+1}^{T} \mathcal{G}_\tau(\pi|Q^*) \\
\mathcal{G}_\tau(\pi|Q^*) &= \mathbb{E}_{Q^*(o_\tau,s_\tau|\pi)}\left[\log Q^*(s_\tau|\pi) - \log \tilde{P}(o_\tau, s_\tau)\right] \\
&= \underbrace{D_{KL}\left[Q^*(s_\tau|\pi)||\tilde{P}(s_\tau)\right]}_{\text{Risk}} + \underbrace{\mathbb{E}_{Q^*(s_\tau|\pi)}\mathcal{H}\left[\tilde{P}(o_\tau|s_\tau)\right]}_{\text{Ambiguity}} \\
&= \mathbb{E}_{Q^*(o_\tau|\pi)}\left[-\log \tilde{P}(o_\tau)\right] + \mathbb{E}_{Q^*(o_\tau|\pi)} D_{KL}\left[Q^*(s_\tau|o_\tau,\pi)||\tilde{P}(s_\tau|o_\tau)\right] \\
&\quad - \mathbb{E}_{Q^*(o_\tau|\pi)} D_{KL}\left[Q^*(s_\tau|o_\tau,\pi)||Q^*(s_\tau|\pi)\right] \\
&\geq \underbrace{\mathbb{E}_{Q^*(o_\tau|\pi)}\left[-\log \tilde{P}(o_\tau)\right]}_{\text{Expected value}} - \underbrace{\mathbb{E}_{Q^*(o_\tau|\pi)} D_{KL}\left[Q^*(s_\tau|o_\tau,\pi)||Q^*(s_\tau|\pi)\right]}_{\text{Expected information gain}}
\end{aligned}
\tag{2.43}
$$

where $\mathcal{H}[\cdot]$ denotes Shannon entropy. The second line shows that the EFE can be decomposed into a KL divergence between the predictive and desired distribution — a measure of risk — and an expected observation entropy — a measure of uncertainty. The last line shows that the EFE is an upper bound on the negative expected desired distribution likelihood — a measure of value — and the negative expected KL divergence between a posterior and a prior — a measure of information gain.

The decomposition above is viewed as a central characteristic of active inference, equipping the agent with an ability to handle epistemic uncertainty of hidden states [67, 18, 30]. The ambiguity and information gain terms promote visiting states that lead to uncertainty reduction, or increase in belief precision. In practice, active inference agents show greater propensity for exploration and faster adaptation in changing environments [68, 19, 69].

Despite having an attractive interpretation as optimal handeling of exploration and exploitation, the EFE objective is often questioned for its origin, motivation, and consistency with the FEP. This is usually supported by a *reductio ad absurdum* argument that an agent whose goal is to minimize

free energy must be endowed with a prior belief as such [67]. This argument is met by a handful of objections, attempts for better unification with the FEP, and proposals of alternative objectives, e.g., [70, 31, 71, 72]. I give a novel justification for EFE in the next section (2.4.3) and a brief overview of other debates and proposals in section 2.4.5.

2.4.3 Connecting Expected Free Energy and Expected Value

This section aims to establish a connection between active inference and the expected value framework in traditional POMDP solution methods using a novel derivation of the EFE function. The connection is presented in the following proposition.

Proposition 1. *Active inference optimizes hidden state information gain and the following reward function:*

$$R(b_t, a_t) = \mathbb{E}_{P(o_{t+1}|b_t,a_t)}\left[\log \tilde{P}(o_{t+1}|b_t, a_t)\right] \tag{2.44}$$

The derivation starts with the premise of the FEP that agent behavior is governed by the drive to maximize the expected future model evidence, i.e., the expected log marginal likelihood, given history h_t defined as:

$$\mathcal{L}(h_t) = \mathbb{E}_{\substack{P(o_{t+1:t+H}|ha_{t:t+H-1}) \\ \pi(a_{t:t+H-1}|h_{t:t+H-1})}}\left[\log \tilde{P}(o_{t+1:t+H}|ha_{t:t+H-1})\right] \tag{2.45}$$

where t denotes the current lifetime. Here, P and $\tilde{P}$ are two models defined on the same space. P is a predictive model of the environment dynamics and $\tilde{P}$ is an evaluative model scoring the desirability of a trajectory $o_{t+1:t+H}$. Both models contain latent variables s with a single time slice defined according to the POMDP structure:

$$P(o_{t+1}|h_t, a_t) \triangleq \sum_{s_{t+1}} P(o_{t+1}|s_{t+1})P(s_{t+1}|h_t, a_t) \tag{2.46}$$

It is immediate that (2.45) is a special case of the POMDP objective defined in (2.2) with a

reward function defined as the log marginal likelihood of a trajectory under the evaluative model.

For any given o_{t+1}, we can show that the model evidence is equal to:

$$\begin{aligned}
&\log \tilde{P}(o_{t+1}|h_t, a_t) \\
&= \mathbb{E}_{\tilde{P}(s_{t+1}|h_t,a_t,o_{t+1})}\left[\log \tilde{P}(o_{t+1}, s_{t+1}|h_t, a_t) - \log \tilde{P}(s_{t+1}|h_t, a_t, o_{t+1})\right] \\
&= \mathbb{E}_{\tilde{P}(s_{t+1}|h_t,a_t,o_{t+1})}\Big[\log \tilde{P}(o_{t+1}|s_{t+1}) + \log \tilde{P}(s_{t+1}|h_t, a_t) - \log \tilde{P}(s_{t+1}|h_t, a_t, o_{t+1})\Big] \\
&= -D_{KL}\left[\tilde{P}(s_{t+1}|h_t, a_t, o_{t+1})||\tilde{P}(s_{t+1}|h_t, a_t)\right] + \mathbb{E}_{\tilde{P}(s_{t+1}|h_t,a_t,o_{t+1})}\left[\log \tilde{P}(o_{t+1}|s_{t+1})\right]
\end{aligned} \tag{2.47}$$

We will now assume the agent is in equilibrium with the environment such that its predicted trajectory matches the desired trajectory, i.e., $P = \tilde{P}$. This allows us to mix the two distributions, arriving at the following form of expected model evidence:

$$\begin{aligned}
&\mathbb{E}_{P(o_{t+1}|h_t,a_t)}\left[\log \tilde{P}(o_{t+1}|h_t, a_t)\right] \\
&= -\mathbb{E}_{P(o_{t+1}|h_t,a_t)} D_{KL}\left[P(s_{t+1}|h_t, a_t, o_{t+1})||\tilde{P}(s_{t+1}|h_t, a_t)\right] - \mathbb{E}_{P(s_{t+1}|h_t,a_t)}\mathcal{H}\left[\tilde{P}(o_{t+1}|s_{t+1})\right]
\end{aligned} \tag{2.48}$$

where we have used the relationship $P(s|o,\cdot)P(o|\cdot) = P(o|s)P(s|\cdot)$ to rewrite the second term as an expected entropy.

Lastly, we will add an expected information gain term to the expected model evidence defined as:

$$\begin{aligned}
\mathcal{I}(o_{t+1}, s_{t+1}|h_t, a_t) &\triangleq \mathbb{E}_{P(o_{t+1}|h_t,a_t)} D_{KL}\left[P(s_{t+1}|h_t, a_t, o_{t+1})||P(s_{t+1})\right] \\
&= \mathbb{E}_{P(o_{t+1},s_{t+1}|h_t,a_t)}\left[\log P(s_{t+1}|h_t, a_t, o_{t+1}) - \log P(s_{t+1}|h_t, a_t)\right]
\end{aligned} \tag{2.49}$$

The result is equivalent to the single-step EFE:

$$\begin{aligned}
&\mathbb{E}_{P(o_{t+1}|h_t,a_t)}\left[\log \tilde{P}(o_{t+1}|h_t,a_t)\right] + \mathcal{I}(o_{t+1}, s_{t+1}|h_t,a_t) \\
&= -D_{KL}\left[P(s_{t+1}|h_t,a_t)||\tilde{P}(s_{t+1}|h_t,a_t)\right] - \mathbb{E}_{P(s_{t+1}|h_t,a_t)}\mathcal{H}\left[\tilde{P}(o_{t+1}|s_{t+1})\right] \\
&= -\mathcal{G}(h_t,a_t)
\end{aligned} \tag{2.50}$$

This derivation helped clarifying two important assumptions in active inference:

1. The agent is in equilibrium with the environment such that variables between the predictive and evaluative distributions can be mixed.

2. The agent maximizes expected information gain in addition to expected model evidence.

One interpretation of the equilibrium assumption is an optimistic prior belief of behaving optimally in the future. This creates a connection to control-as-inference (see Section 2.3.4.1). To what extent the agent can achieve equilibrium likely depends on the actual environment and is currently an unresolved question [73, 74, 75]. Separately, the addition of expected information gain seems to undermine the principled motivation for maximizing expected model evidence. Indeed, expected model evidence already admits a risk-ambiguity decomposition as shown in (2.48). Given the entropy maximizing property of reverse KL divergence [76], expected model evidence also encourages exploratory behavior by covering a larger state space. The introduction of expected information gain is likely related to conflating the semantics of variational distributions. This has been discussed in [77] and studies including [78, 79] have cited the disappearance of exploratory behavior when specific model class or inference methods are used.

Viewing the EFE as a reward function and viewing the EFE prior as a prior over optimal actions has both theoretical and practical benefits. It provides a justification for the heuristic motivation of the EFE prior in (2.39) based on a principled motivation for maximizing expected model evidence — the central claim of the FEP. This allows active inference to adopt optimization methods developed by the planning and reinforcement learning communities (see Section 2.2 and 2.3) and scale to complex environments. For a comparison of the active inference objective developed in this

section, labeled as *exact active inference*, and the *variational active inference* objectives presented in Section 2.4.2, see Table 2.1.

Table 2.1: Perception and action update rules in variational and exact active inference

Perception	Variational	$Q^*(s_\tau\|\pi) \propto \exp\left(\mathbb{E}_{Q^*(s_{\tau-1}\|\pi)}\left[\log P(o_\tau, s_\tau\|s_{\tau-1}, \pi)\right] + c\right)$
	Exact	$b(s_\tau\|a_{\tau-1}) \propto \exp\left(\log P(o_\tau, s_\tau\|b_{\tau-1}, a_{\tau-1})\right)$
Action	Variational	$\mathcal{G}(\pi\|Q^*) = \mathbb{E}_{Q^*}\left[\sum_{\tau=t+1}^{t+H} \log Q^*(s_\tau) - \log \tilde{P}(o_\tau, s_\tau)\right]$
	Exact	$G(b_t, a_t) = \mathbb{E}_P\left[\sum_{\tau=t+1}^{t+H} \log \sum_{s_\tau} \tilde{P}(o_\tau, s_\tau)\right]$

2.4.4 Scaling Active Inference

Scalability is a central challenge in active inference. Given that the agent represents the complete sequence of states and actions, extension to settings with extended lifetime is difficult. To this end, the majority of active inference implementations have adopted two modifications: 1) perform state estimation of the most recent state without retrospective estimation, and 2) represent policies instead of the complete action sequence [69, 80, 81].

Under this modification, the agent generative model is factorized as:

$$P(o_{1:t}, s_{1:t}, a_{1:t}) = \prod_{\tau=1}^{t} P(o_\tau|s_\tau)P(s_\tau|s_{\tau-1}, a_{\tau-1})P(a_\tau|s_\tau) \tag{2.51}$$

where $P(a_\tau|s_\tau) = 1, \forall \tau \leq t-1$. Using the same free energy minimization method for perception as in Section 2.4.2.1, we can show that the optimal variational distributions correspond to Bayesian beliefs without additional effect from representing future states.

The action prior is defined as:

$$P(a_t|s_t) \propto \exp\left(-\mathcal{G}(s_t, a_t)\right) \tag{2.52}$$

where:

$$\mathcal{G}(s,a) \triangleq \mathbb{E}_{P(o',s'|s,a)}\left[\log P(s'|s,a) - \log \tilde{P}(o',s')\right] + \log P(a|s) + \max_{P(a'|s')} \mathbb{E}_{P(s',a'|s,a)}\left[\mathcal{G}(s',a')\right] \tag{2.53}$$

It is easy to see that the action posterior $Q^*(a_t)$ equals the prior.

(2.53) can be seen as a QMDP approximation of the POMDP formulation of active inference introduced in Section 2.4.3. However, different from QMDP agents, observation entropy is now included in the reward function. An agent optimizing this reward will seek states with low observation entropy, and thus likely maintain high belief precision throughout its lifetime, which makes QMDP's full future observability assumption valid. However, whether the agent will actually maintain high belief precision in the future depends on the property of the actual POMDP and the agent's environment model. To equip agents with better handling of uncertainty, more recent versions of active inference define the EFE prior in terms of current and future beliefs rather than states [82], which becomes equivalent to the formulation in Section 2.4.3.

2.4.5 Neuroscience Motivation and Agent Objective Design

In active inference, subsuming actions into the prior is motivated by its predictive processing root. Traditionally, human motor behavior is modeled with decoupled perception and control systems similar to the MBRL architecture [23]. The communication between perception and control requires the controller to send control commands via an efference copy to the perception system in order to enable accurate state estimation. However, contemporary motor neuroscience suggests that humans do not represent self-generated actions in the form of motor commands and efference copies, as it would be otherwise difficult to explain phenomena such as sensory attenuation and the complex interaction between self-tickling and attention [83, 84, 85]. Instead, humans predict the consequences of desired actions and have the prediction error resolved by reflex, in turn arriving at the desired state. As such, Friston and colleagues argue that value is not the cause of movements but rather the consequence and has advocated for optimal control without cost function [83, 86].

In the Bayesian framework, prediction can be understood as the formation of expectation mediated by the prior distribution and prediction error is reduced by updating the prior to the posterior for passive Bayesian reasoners [87, 88]. This is however undesirable for active agents since their goals cannot be realized through only state estimation. Active inference's resolution is to equip the agent with a strong and biased prior such that it cannot be overridden by state estimation. In particular, it does so by equipping agents with beliefs about state transitions controlled by an optimal policy in earlier implementations, e.g., [83, 89], and in the version presented in this chapter, it equips the agent with higher prior probabilities on optimal state-action trajectories.

Such a unification is controversial among cognitive scientists and philosophers with both supporters [90, 91, 92, 93, 94, 95, 24] and objectors [71, 96, 97, 33, 98, 70]. Drawn by the efference copy argument, the majority of supporters are motivated by the embodied cognition principle that perception should serve actions rather than being a mirror of the environment [24]. Baltieri and Buckley showed that by not representing actions, active inference excels in control tasks when unknown external forces are applied, whereas the decoupled system fails due to inaccurate state estimation resulting from the mis-specified model [99]. Objectors are concerned with the triviality of the unification. Gottwald and Braun [70] illustrated the inconsistency between the joint optimization framework claimed by active inference supporters and the actual formulations and implementations since 2010.

Alternatively, the community has tried to propose agent objectives that are potentially more consistent with the FEP. In [72], Parr and Friston proposed to remove the EFE action prior with a reward factor added to the generative model such that free energy minimization can be done without decoupling perception and action. As argued by Gottwald and Braun [70], this modification makes the active inference formulation virtually equivalent to control-as-inference. Millidge et al. [31] proposed to replace EFE with a planning objective called the Free Energy of Expected Future (FEEF), defined as the KL divergence between the predicted and desired *joint* state-observation distributions. Hafner et al. [100] proposed minimizing the joint latent-observable distribution as a general principle for actions and perception.

Despite the extensive effort towards either rationalizing active inference or proposing alternative objectives, currently there is no consensus on how perception and action should be jointly optimized in autonomous agents.

2.5 Summary

In this chapter, I reviewed POMDP as a minimal perception-action loop and traditional approaches to autonomous agent design, including both learning and planning, revealing a need for the unification of perception and action. I then provided an extensive review of active inference as a candidate framework for unified perception and action. The review contained the most up-to-date active inference formulations, its neuroscience motivations, and current debates regarding the validity and novelty of this framework. To connect active inference with traditional agent design frameworks, I presented a novel derivation of the expected free energy — a central quantity governing active inference agent's goal-directed and information-seeking behavior — based on the FEP, thus justifying its design choice. This novel connection and the discussions in section 2.4.5 show that while active inference represents one attempt to unify perception and action, the objective mismatch problem is still far from being resolved. Equipped with this insight, the rest of the dissertation aims to understand the coupling between perception and action in human agents, in turn shedding lights on novel objective design for synthetic agents.

3. BEHAVIOR UNDERSTANDING AS THEORY OF MIND INFERENCE

3.1 Introduction

The last chapter reviewed models of autonomous agents and revealed a need for understanding the coupled roles of perception and action in developing and generating agent behavior. This chapter introduces Theory of Mind (TOM) inference as a framework for behavior understanding in autonomous agents. Usually set with an agent observing another agent, referred to as the target, TOM formulates the agent's perceptual process as interpreting the beliefs and desires of the target, where the beliefs and desires are related by an axiom of rationality [101]. Desire inference have been studied extensively in control, robotics, machine learning, economics, among other areas, under the titles of inverse optimal control, inverse reinforcement learning, structural estimation, etc. [102, 103, 104], where the goal of the observer is to infer the desired state of the world pursued by the target. However, belief inference is as fundamental as desire inference: children as young as 4 years old show understanding of false beliefs [10]. The combination of belief and desire enriches the set of behavior expressible by an agent [11].

While simultaneous belief-desire inference is attractive by providing better characterization of target behavior, it creates a difficult inference problem because the attribution of belief and desire to a set of observed behavior is typically not unique — there exists alternative belief-desire pairs that explain the observed behavior equally well. For example, when a mouse does not take a certain turn in a maze, it is not clear whether it beliefs the turn is blocked or it desires an alternative route. While a Bayesian approach to the TOM problem can potentially address the non-uniqueness problem by encoding the uncertainty in the posterior distribution [105], a posterior that is too uncertain prohibits the observer from making precise decisions. The main cause of the non-uniqueness problem in TOM, I propose, is a lack of structure in the inference problem. Typically, no assumption is made about the target's belief or desire, i.e., the target is allowed to take on any belief or desire, and no specification is made about the relationship between the observer, the target, and the envi-

ronment. As pointed out by Jara-Ettinger [14], human TOM inference is far from uninformative and highly structured, e.g., they do not use uniform prior or make inference about things that are apparently true. I propose to address the non-uniqueness of TOM inference by equipping the observer with a structured prior on the likely configurations of target belief and desire. Importantly, the prior should be based on and consistent with the data observed rather than hard coded by a human designer.

In what follows, I start by formalizing the TOM inference problem in a Bayesian framework, clarifying the relationship between the target, the environment, and the observer. I then review existing approaches to desire and belief inference. Following the review, I discuss the uniqueness problem, relating it to generative vs. discriminative modeling approaches in machine learning. Lastly, I propose a method to learn structured priors from empirical data leveraging the idea that target agents are developed by learning about the shared environment.

3.2 Bayesian Theory of Mind

In the Bayesian Theory of Mind (BTOM) inference setting [105], we have an observer agent watching a target agent interacting with an environment. The target agent and the environment exchange information via observable signals $o \in \mathcal{O}$ generated by the environment and actions $a \in \mathcal{A}$ responded by the agent for a finite number of time steps $T < \infty$. As a result, the observer receives a finite sequence of observation-action pairs: $\tau = \{o_{1:T}, a_{1:T}\}$. The target agent has a configuration, defined by a set of parameters θ, which gives rise to its beliefs and desires at different time steps. In other words, knowing the target agent's parameters θ allows the observer to uniquely infer its beliefs and desires at any time step given the interaction history $h_t = \{o_{1:t}, a_{1:t-1}\}$. None of the above requires the target parameters θ to be equal to the environment parameters ϕ. In the most general case, both the target and the environment generate signals based on the entire interaction history:

$$o_t \sim P(o_t|h_{t-1}, a_{t-1}; \phi), \quad a_t \sim P(a_t|h_t; \theta) \tag{3.1}$$

Alternatively, the observer may assume the environment is a POMDP and the target holds the same belief. An illustration of this process for a POMDP environment is shown in Fig. 3.1.

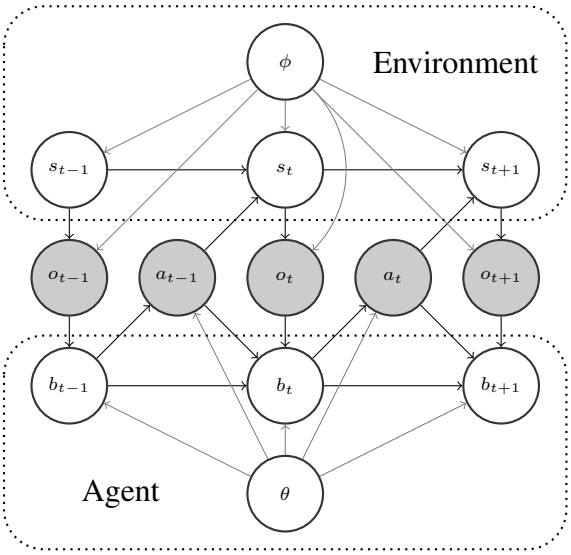


Figure 3.1: Bayesian network of BTOM. Observable nodes by both agents are colored in gray and unobservable nodes are transparent. Environment parameters ϕ generate environment states and observations. Agent parameters θ generate agent beliefs and actions.

As a Bayesian agent, the observer has a prior belief over the target agent's parameters $P(\theta)$. BTOM corresponds to finding the posterior belief over the target agent's parameters upon observing the interaction sequence:

$$\begin{aligned} P(\theta|o_{1:T}, a_{1:T}) &= \frac{P(o_{1:T}, a_{1:T}|\theta)P(\theta)}{\int_\theta P(o_{1:T}, a_{1:T}|\theta)P(\theta)} \\ &= \frac{P(o_{1:T}|h_{1:T-1}, a_{1:T-1}; \phi)P(a_{1:T}|h_{1:T}; \theta)P(\theta)}{\int_\theta P(o_{1:T}|h_{1:T-1}, a_{1:T-1}; \phi)P(a_{1:T}|h_{1:T}; \theta)P(\theta)} \\ &= \frac{P(a_{1:T}|h_{1:T}; \theta)P(\theta)}{\int_\theta P(a_{1:T}|h_{1:T}; \theta)P(\theta)} \\ &= \frac{\prod_{t=1}^{T} P(a_t|h_t; \theta)P(\theta)}{\int_\theta \prod_{t=1}^{T} P(a_t|h_t; \theta)P(\theta)} \end{aligned} \tag{3.2}$$

The second line reduces to the third line since the observations are not generated by the target parameters θ. The last line shows that each action taken by the target agent is *causally* conditioned

on the current history, i.e., there is no dependency between the target's current and future actions or observations that can influence the observer's belief about θ [106, 107].

We are often interested in a point estimate of the target parameters that is most representative of its behavior. A well motivated choice for the point estimate is the parameters with the highest posterior probability known as the Maximum A Posterior (MAP) estimate:

$$\begin{aligned}\theta^{MAP} &= \arg\max_{\theta} P(\theta|o_{1:T}, a_{1:T}) \\ &= \arg\max_{\theta} \prod_{t=1}^{T} P(a_t|h_t;\theta)P(\theta) \\ &= \arg\max_{\theta} \sum_{t=1}^{T} \log P(a_t|h_t;\theta) + \log P(\theta)\end{aligned} \tag{3.3}$$

As we will see later, both the likelihood $P(a_t|h_t;\theta)$ and the prior $P(\theta)$ play important but different roles: the likelihood relates observed actions to target parameters that likely generated target behavior and the prior screens off target parameters believed to be unlikely a priori. When no assumption is made about the target agent, i.e., a uniform prior is used, the MAP estimate is equivalent to the maximum likelihood estimate (MLE).

3.2.1 BTOM of POMDP Agents

This section instantiates BTOM of a target agent with a POMDP model of the environment. This amounts to specifying the structure of the likelihood function in (3.2) according to the variable dependencies implied by the POMDP structure shown in Fig. 3.1. A POMDP agent has beliefs about the environment configuration $b(\phi_t)$ and beliefs about the environment state $b(s_t)$ at every time step. The agent also has desire over the state of the environment and actions to pursue defined by the reward function $R(s_t, a_t)$. We denote the total target configuration with $\theta = \{\theta_1, \theta_2\}$ corresponding to the parameters of target belief and desire. Here, we make a simplifying assumption that the target's belief about the environment configuration is a point estimate which does not change over the course of interaction with the environment: $b(\phi_t) = \delta(\phi_t - \tilde{\phi})$. This is a reasonable assumption when the duration of interaction is short. We emphasize that the point belief $\tilde{\phi}$ is *not*

necessarily equal to the true environment parameters ϕ.

The action likelihood of a POMDP agent is:

$$P(a_{1:T}|h_{1:T};\theta) = \prod_{t=1}^{T} \pi(a_t|b_t;\theta)\delta\left(b_t - b(s_t|h_t;\theta_1)\right) \tag{3.4}$$

where π is governed by an axiom of rationality, e.g., one of the planning algorithms introduced in Chapter 2.

MLE BTOM inference corresponds to finding θ that maximizes (3.4). This requires inverting the target's planning process, i.e., the process of finding the policy π. This is a challenging problem since the planning process itself is challenging especially when the dynamics is complex. The next section reviews existing TOM algorithms.

3.3 Review of TOM Inference Algorithms and Applications

Prior work on TOM inference can be straightforwardly categorized into desire inference and belief or joint belief-desire inference. Desire inference has been studied extensively in fields such as computer science, economics, and psychology under the titles of inverse reinforcement learning, inverse optimal control, inverse decision theory, structural estimation, etc., [102, 103, 14, 104]. In inverse reinforcement learning and inverse optimal control, there is also an emphasis on reverse engineering a policy for task-solving, which interacts with the requirement of inference accuracy but has more nuanced practical implications. In contrast, belief inference has received little attention. In both categories, the majority of studies assume agent beliefs about the environment are given and fixed, or its beliefs are perfect copies of the true environment, with a few notable exceptions [108, 109, 105]. I will thus start by reviewing the most relevant work in inverse reinforcement learning and then review belief inference with a focus on the diversity of approaches and the identifiability or uniqueness of belief inference. I end with a review of the practical use cases of TOM inference.

3.3.1 Desire Inference with Inverse Reinforcement Learning

Inverse reinforcement learning (IRL) is a particular method for inverse optimal control or inverse decision theory, where the planning process is performed approximately by reinforcement learning. The goal of IRL is to recover the reward function optimized by the target agent. IRL typically assumes the target agent has a fixed belief about the environment that exactly mirrors the true environment [110]. This belief can be represented by a set of matrices in a discrete environment, a set of equations in a continuous environment, or a simulator for a highly complex environment. Although not strictly required, IRL typically assumes the environment is fully observable, i.e., the environment is treated as an MDP.

Earlier works in IRL are concerned with the degeneracy of the inference problem, where the observed behavior can be explained equally well by a large set of reward functions [102]. This is mainly due to the non-smooth nature of MDPs, where there always exists a deterministic optimal policy [111]. Thus, two reward functions can give rise to the same optimal policy as long as they do not alter the ordering of optimal actions in each state. Notable proposals for addressing the uniqueness problem include the maximum margin formulation, which finds a reward that maximally distinguishes the optimal from the sub-optimal policies [112]. Another challenge is the modeling of noisy behavior, where the observed agent chooses multiple different actions in the same state with different frequencies. The dominant approach is to assume agents choose actions with probability proportional to the exponential of the cumulative rewards. This can be motivated by a noisy reward model [104] or a Bayesian approach [113].

The majority of recent works have settled on the Maximum Causal Entropy (MCE) IRL framework, which simultaneously addresses degeneracy and noisy actions [114]. Although initially motivated by the principle of maximizing the (causal) entropy of a predictive model, i.e., the policy, which matches observed agent behavior while being as parsimonious as possible, MCE-IRL can be equally viewed as inferring the reward function of a planning-as-inference agent which simultaneously maximizes reward and policy entropy (see Section 2.3.4.1, [115]). The policy of such an agent is always smooth in the reward functions, and the mapping from policy back to re-

ward is unique up to a constant [116]. Recent works have extended MCE-IRL to high dimensional environments with applications in robotics and autonomous driving [117, 118, 119].

3.3.2 Belief Inference Frameworks

This section reviews belief inference and joint belief-desire inference algorithms with a focus on the following two aspects: 1) assumptions on the subjectivity of beliefs, and 2) the specific inference algorithm. There are extensions of IRL which perform belief-desire inference in a decoupled fashion [120, 121, 122]. They estimate the environment parameters in the first stage and fix the estimated parameters while inferring agent reward in the second stage. These methods are not considered belief inference in the BTOM context, since these beliefs are not scored by the likelihood function in the BTOM definition in (3.2). In other words, they represent the observer's belief about the environment but not the target's belief.

Among the BTOM approaches, most belief or joint inference methods assume the target agent models the environment as an MDP [123, 124, 125, 126, 109, 127] or POMDP [108, 128, 129, 105, 130]. [125] and [109] assume the reward function is given and only estimate the agent's model of the environment, while the rest of the works mentioned perform simultaneous estimation of belief and desire. Most works make explicit assumptions about the environment model family as either discrete [124, 125, 108, 105] or linear-Gaussian [109, 129, 128], with an exception proposed by Gangwani et al. [130], where agent beliefs are parameterized by a neural network. In the context of inferring human's biased beliefs about the environment, Reddy et al. [109] and Shah et al. [131] raised an important question that without further constraints on the environment model parameters, i.e., beyond the fact that it is parameterized by a specific model class, the model is empirically observed to be unidentifiable, i.e., different solutions explain the target agent equally well. They both proposed to regularize the belief model to make accurate predictions on observed data. Other works such as [127, 126, 105] propose to capture parameter unidentifiability with Bayesian approaches.

These works have also presented a variety of approaches in addressing the challenging inverse planning problem, although all of them used the common constraint that agent policies should

respect the Bellman equation. Bacon et al. [123] and Reddy et al. [109] proposed to directly solve the constrained optimization problem using either the Lagrangian dual descent-ascent algorithm or the penalty method. Herman et al. [124], Golub et al. [125], and Wu et al. [108] proposed to directly differentiate the Bellman equation. Their derivations show that the belief gradient can be expressed as the gradient of the expected value, implying potential cause for non-identifiability. The method of Wu et al. [108] is complicated by the fact that the target agent is assumed to have a POMDP model. The authors propose to discretize the belief space and infer the target belief about the environment state at different time steps using a message-passing algorithm. Baker et al. [105] avoided direct optimization with Markov Chain Monte Carlo sampling. Kwon et al. [128] used meta reinforcement learning to train an agent on all configurations of the environment and subsequently used this agent to generate the beliefs and action probabilities required by the message-passing algorithm similar to Wu et al. [108]. Gangwani et al. [130] performed density matching on beliefs using the adversarial imitation learning algorithm [132].

The reviewed studies show that, unlike desire inference, there is currently no consensus on the best algorithmic framework for belief inference. The joint belief-desire inference setting is further complicated by the fact that desire inference has to be based on an intermediate belief inference result, while the update direction for target beliefs also depends on the inferred target desire. Such a coupling implies potential non-identifiability, which is often not cited in the literature. A likely reason for the lack of reference to non-identifiability is that the majority of work operates in small environments with an extensive amount of prior knowledge injected to constrain the hypothesis space of target beliefs. However, these restrictions are not desirable for the application of TOM inference in general environments. Thus, there is substantial value in clarifying TOM identifiability and requirements on the type of environments and target behavior.

3.3.3 The Usefulness of TOM Inference

The purpose of TOM inference is to provide better characterization of agent beliefs and desire than assuming agent beliefs coincide with the true environment or an estimate thereof in decoupled approaches. There are two main use cases of TOM inference. The first case uses TOM as

a research tool for understanding human cognitive behavior in psychology experiments [11, 133] and mapping inferred mental states to neural correlates or other behavioral markers [134, 135]. Equipped with this knowledge, a robot can provide assistance to humans, or other target agents, by augmenting their perception or control. For example, Reddy et al. [109] inferred human participants' beliefs about the environment dynamics in a rocket-landing game assuming their desire equals to the actual game reward. The inferred beliefs show that humans perceive the game to be slower than the actual game speed. By augmenting participant actions towards actions leading to the intended states at the lower believed speed, they achieved a significant increase in the task completion rate.

The second and less explored use case is in learning a model of the environment without interaction by extracting knowledge from the target agent [136]. This is most useful when the observer has limited access to an environment due to excessive risk and has to estimate a model of the environment from historical data and then plan a policy from the estimated environment model, i.e., model-based offline IRL. A maximum likelihood estimate of the environment parameters will make inaccurate predictions in states that do not exist in the dataset, e.g., uniform prediction in the tabular model representation, and thus plan a suboptimal policy. In contrast, joint inference allows target agent decisions to inform the estimation of environment parameters. In this way, the environment model becomes task-aware and avoids issues associated with the model-policy objective mismatch in MBRL discussed in Section 2.3.3.

3.4 The Uniqueness of BTOM Inference

The non-unique nature of BTOM is intuitive: agent behavior can be motivated by either desire or beliefs about what is possible or impossible in the environment. However, it is far less clear to what extent it depends on the environment, behavior observed, or assumptions and priors on the agent's configuration. The next two sections study BTOM identifiability assuming MDP and POMDP environment models. I show that belief and joint inference in both settings are not identifiable as a result of under-determined systems.

3.4.1 (Un)idenfifiability of MDP Models

We start by studying the identifiability of desire, belief, and joint inference problems assuming the target agent models the environment as a fully observable MDP. We make no additional assumption other than the target agent being an optimal planner with respect to (w.r.t.) its model of the environment. For the desire or belief inference case, we assume the target belief (i.e., the environment model) or desire (i.e., the reward function) is known. For the joint inference case, we assume both are unknown. We will show that only desire inference is identifiable while the other two cases are in general unidentifiable. In the joint inference case, we can find complementary desires to compensate for changes in beliefs without changing subsequent behavior. In the belief inference case, I show that the unidentifiability is due to an under-determined system. In each case, I state the result first and then provide the analysis.

Proposition 2. ***(MDP joint inference)*** *Joint belief-desire inference in the MDP setting is in general unidentifiable.*

We start by considering the joint belief-desire inference setting. The target agent behavior is generated from a policy $\pi(a_t|s_t)$ planned in an MDP with *subjective* transition model $\tilde{P}(s_{t+1}|s_t, a_t)$ and reward function $R(s_t, a_t)$. To simplify the analysis, we assume the policy is parameterized by a Q function which is given or can be estimated accurately from target behavior (e.g., using Energy Based Models [137, 138, 139]). The Q and value function associated with the policy satisfy the Bellman equation written in matrix form as:

$$\mathbf{Q} = \mathbf{R} + \mathbf{PV} \tag{3.5}$$

Let us introduce a pair of alternative reward function $R'(s_t, a_t)$ and transition model $\tilde{P}'(s_{t+1}|s_t, a_t)$ while fixing the Q and value function so that the optimal policy stays the same. The difference

between the old and new reward functions can be written as:

$$\begin{aligned}\Delta\mathbf{R} &= \mathbf{R}' - \mathbf{R} \\ &= (\mathbf{Q} - \mathbf{Q}) - (\mathbf{P}'\mathbf{V} - \mathbf{P}\mathbf{V}) \\ &= -\Delta\mathbf{P}\mathbf{V}\end{aligned} \tag{3.6}$$

This shows that for an arbitrary adjustment in the *subjective* transition probabilities, we can always find a corresponding adjustment in the reward function that keeps the value function and thus the policy unchanged. The adjustment in reward is the negative change in expected value.

Proposition 3. ***(MDP desire inference)*** *Desire inference in the MDP setting is identifiable.*

The above joint inference analysis also shows that for a fixed transition model (i.e., $\Delta\mathbf{P} = \mathbf{0}$ where $\mathbf{0}$ is a zero matrix), the reward function is uniquely determined (i.e., $\Delta\mathbf{R} = \mathbf{0}$) if the value functions associated with the policy are given. Thus, target desire can be identified from observed behavior following the possibility of identifying Q functions [137].

Proposition 4. ***(MDP belief inference)*** *Belief inference in the MDP setting is in general unidentifiable.*

To study the identifiability of belief inference, we fix the target reward function by setting $\Delta\mathbf{R} = \mathbf{0}$. The identifiability problem is translated into finding a different transition model $\mathbf{P}'$ such that:

$$\Delta\mathbf{P}\mathbf{V} = \mathbf{0} \tag{3.7}$$

$\Delta\mathbf{P}$ can be expressed as:

$$\begin{aligned}\Delta\mathbf{P} &= \mathbf{0}\mathbf{V}^\dagger + \mathbf{Z}(\mathbf{I} - \mathbf{V}\mathbf{V}^\dagger) \\ &= \mathbf{Z}(\mathbf{I} - \mathbf{V}\mathbf{V}^\dagger)\end{aligned} \tag{3.8}$$

where $\mathbf{Z} \in \mathbb{R}^{|S||A| \times |S|}$ is an arbitrary matrix of the same size as $\Delta\mathbf{P}$ and $\mathbf{V}^\dagger = \mathbf{V}^T / ||\mathbf{V}||_2$ [140]. This can be rewritten as a set of systems of linear equations, one for each source state and action:

$$(\mathbf{I} - \mathbf{V}\mathbf{V}^\dagger)^T \mathbf{Z}[i] = \Delta\mathbf{P}[i] \tag{3.9}$$

where $\mathbf{X}[i]$ is the i^{th} row of the matrix $\mathbf{X}$ with $i \in \mathbb{Z}^{|S||A|}$. The system of equations has a solution $\mathbf{Z}[i]^*$ iff $(\mathbf{I} - \mathbf{V}\mathbf{V}^\dagger)^T$ is invertible, which requires that the nonzero eigenvalue of $\mathbf{V}\mathbf{V}^\dagger$: $\mathbf{V}^\dagger\mathbf{V} \neq -1$. This is true since all elements of $\mathbf{V}^\dagger\mathbf{V}$ are positive.

The result of this analysis is that for a given transition-reward pair, we can usually choose an arbitrary matrix $\mathbf{Z}$ and obtain a new transition matrix $\mathbf{P}'$ (subject to the constraint that $\mathbf{P}'$ is a stochastic matrix) as:

$$\mathbf{P}' = \mathbf{P} + \mathbf{Z}(\mathbf{I} - \mathbf{V}\mathbf{V}^\dagger) \tag{3.10}$$

which leaves the value function and the policy unchanged. Thus, even in the case where the target agent's environment model is fully observable and the desire is specified, belief inference is still unidentifiable. This supports the empirical observations made in [109, 131] and challenges other belief and joint belief-desire inference methods in MDPs [124].

3.4.2 (Un)identifiability of POMDP Models

Analyzing the identifiability when the agent environment model is partially observable is more challenging than when the agent environment model is fully observable. This is because the model parameters affect not only policy planning but also the agent's belief about the environment at each time step. We will approach the analysis using a notion of *observational equivalence*.

Specifically, we will consider the problem of whether there exists more than one set of agent parameters θ such that an observer cannot distinguish the data generated from different sets of paramters. We formally define observational equivalence as fixing the marginal distribution of a

finite observation sequence for different parameters:

$$P(o_{1:T}, a_{1:T}) = \sum_{s_{1:T}, b_{1:T}} \prod_{t=1}^{T} P(o_t|s_t)P(s_t|s_{t-1}, a_{t-1})P(b_t|b_{t-1}, a_{t-1}, o_t)P(a_t|b_t) \tag{3.11}$$

where we have omitted the dependence of the belief transition and policy on agent parameters θ for notational clarity.

To achieve observational equivalence, it is sufficient to consider a single time slice shown in Fig. 3.2 with marginal distribution:

$$\begin{aligned} P(o_t, a_t, o_{t+1}) = \sum_{s_t, s_{t+1}} & P(s_t)P(o_t|s_t)P(s_{t+1}|s_t, a_t)P(o_{t+1}|s_{t+1}) \\ & \delta(b_t - b(s_t|b_{t-1}, a_{t-1}, o_t))P(a_t|b_t)\delta(b_{t+1} - b(s_{t+1}|b_t, a_t, o_{t+1})) \end{aligned} \tag{3.12}$$

Since the environment state transition and observation do not depend on agent parameters θ, we only require alternative sets of parameters θ to yield the same distribution of action a_t (i.e., *policy equivalence*) and the posterior belief distributions b_t and b_{t+1} (i.e., *belief equivalence*) for the current time slice. The posterior belief distribution ensures that the prior belief distribution for the next time slice is fixed.

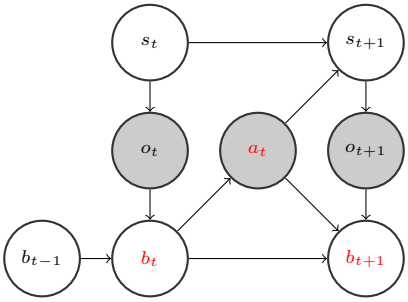


Figure 3.2: A slice of POMDP's dynamic Bayesian network. Observable variables are colored in gray nodes and hidden variables are transparent. For the analysis of observational equivalence, our goal is to find different parameters θ such that the variables in red remain fixed.

We will study the QMDP policy class (see Section 2.2.2.2) for which the Q function can be

expressed mostly as a linear function of agent parameters. We will denote the QMDP Q function as:

$$\begin{aligned} Q(o_t, a_t) &= \sum_{s_t} b(s_t|o_t) Q(s_t, a_t) \\ &= \sum_{s_t} b(s_t|o_t) \left[R(s_t, a_t) + \sum_{s_{t+1}} P(s_{t+1}|s_t, a_t) V(s_{t+1}) \right] \end{aligned} \tag{3.13}$$

with $b(s_t|o_t)$ defined as

$$b(s_t|o_t) = \frac{P(o_t|s_t) \sum_{s_{t-1}} P(s_t|s_{t-1}, a_{t-1}) b(s_{t-1})}{\sum_{s_t} P(o_t|s_t) \sum_{s_{t-1}} P(s_t|s_{t-1}, a_{t-1}) b(s_{t-1})} \tag{3.14}$$

where $b(s_{t-1})$ is assumed to be known and fixed.

Proposition 5. ***(Policy equivalence)*** *There exists more than one policy-equivalent belief-reward pairs for a given QMDP value function.*

Let us denote the matrix form of $b(s_t|o_t)$ with $\mathbf{B}$, where each row corresponds to the posterior belief distribution upon observing o_t. We can write the QMDP Q function in matrix form as:

$$\mathbf{Q} = \mathbf{BR} + \mathbf{BPV} \tag{3.15}$$

We will introduce an alternative set of parameters $\{\mathbf{B}', \mathbf{R}', \mathbf{P}'\}$ and fix the value functions. The change in reward function as a result of changes in the environment parameters can be found as follow:

$$\begin{aligned} \mathbf{B}'\mathbf{R}' &= \mathbf{BR} + (\mathbf{BP} - \mathbf{B}'\mathbf{P}')\mathbf{V} \\ \mathbf{R}' &= \mathbf{B}'^{+} \left[\mathbf{BR} + (\mathbf{BP} - \mathbf{B}'\mathbf{P}')\mathbf{V} \right] \end{aligned} \tag{3.16}$$

Where $\mathbf{B}'^{+}$ denotes $\mathbf{B}'$'s Moore-Penrose pseudo inverse. This equation is an analog of the MDP case (3.6), which shows that for a perturbed set of environment parameters, we can always find a reward function that keeps the Q function and thus the policy unchanged. Thus, if belief equiva-

lence can be achieved, which will be discussed next, we will not be able to tell apart agents with different parameters purely based on observed behavior.

Proposition 6. ***(Belief equivalence)*** *There exists more than one set of belief-equivalent environment model parameters.*

In order for observational equivalence to fully hold, We also need to ensure that alternative sets of parameters do not change the belief distributions at adjacent time steps. Specifically, we require the sequence $\{b_t, b_{t+1}|b_{t-1}, o_t, a_t, o_{t+1}; \theta\}$ to be invariant upon changing θ.

We will first show that for a single time step, the invariance relationship generally holds with different parameters. Let us denote the observation-belief matrix with $\tilde{b}_t$ parameterized by θ such that:

$$\tilde{b}(s_t|o_t) = \frac{P(o_t|s_t;\theta_{11}) \sum_{s_{t-1}} P(s_t|s_{t-1}, a_{t-1};\theta_{12}) b(s_{t-1})}{\sum_{s_t} P(o_t|s_t;\theta_{11}) \sum_{s_{t-1}} P(s_t|s_{t-1}, a_{t-1};\theta_{12}) b(s_{t-1})} \tag{3.17}$$

where θ_{11} and θ_{12} denote the observation and transition distribution parameters, respectively. We will analyze the gradient of these parameters when the KL divergence between b and $\tilde{b}$ is minimized. We can write the KL divergence as:

$$\begin{aligned} D_{KL}\left(b||\tilde{b}\right) &= \mathbb{E}_{b(s_t)}\left[\log b(s_t) - \log \tilde{b}(s_t)\right] \\ &= -\mathbb{E}_{b(s_t)}\left[\log \tilde{b}(s_t)\right] + c \\ &= -\mathbb{E}_{b(s_t)}\left[\log \frac{P(o_t|s_t;\theta_{11}) \sum_{s_{t-1}} P(s_t|s_{t-1}, a_{t-1};\theta_{12}) b(s_{t-1})}{\sum_{s_t} P(o_t|s_t;\theta_{11}) \sum_{s_{t-1}} P(s_t|s_{t-1}, a_{t-1};\theta_{12}) b(s_{t-1})}\right] + c \end{aligned} \tag{3.18}$$

We assume there exists parameters θ_{11} and θ_{12} such that the KL divergence is zero.

The gradient of the KL divergence w.r.t. the parameters are (see Appendix A.2 for derivation):

$$\begin{aligned}
\nabla_{\theta_{11}} D_{KL} &= -\mathbb{E}_{b(s_t)}\left[\frac{\nabla_{\theta_{11}} P(o_t|s_t;\theta_{11})}{P(o_t|s_t;\theta_{11})}\right] \\
&\quad + \frac{1}{Z}\sum_{s_t}\sum_{s_{t-1}} P(s_t|s_{t-1},a_{t-1};\theta_{12}) b(s_{t-1}) \nabla_{\theta_{11}} P(o_t|s_t;\theta_{11}) \\
\nabla_{\theta_{12}} D_{KL} &= -\mathbb{E}_{b(s_t)}\left[\frac{\sum_{s_{t-1}} b(s_{t-1}) \nabla_{\theta_{12}} P(s_t|s_{t-1},a_{t-1};\theta_{12})}{\sum_{s_{t-1}} P(s_t|s_{t-1},a_{t-1};\theta_{12}) b(s_{t-1})}\right] \\
&\quad + \frac{1}{Z}\sum_{s_t}\sum_{s_{t-1}} P(o_t|s_t;\theta_{11}) b(s_{t-1}) \nabla_{\theta_{12}} P(s_t|s_{t-1},a_{t-1};\theta_{12})
\end{aligned} \tag{3.19}$$

where Z is the normalizer in the belief update equation. When the KL divergence is at its minimum, we have:

$$\begin{aligned}
0 &= \nabla_{\theta_{11}} D_{KL} + \nabla_{\theta_{12}} D_{KL} \\
&= -\mathbb{E}_{b(s_t)}\left[\frac{\nabla_{\theta_{11}} P(o_t|s_t;\theta_{11})}{P(o_t|s_t;\theta_{11})} + \frac{\sum_{s_{t-1}} b(s_{t-1}) \nabla_{\theta_{12}} P(s_t|s_{t-1},a_{t-1};\theta_{12})}{\sum_{s_{t-1}} P(s_t|s_{t-1},a_{t-1};\theta_{12}) b(s_{t-1})}\right] \\
&\quad + \frac{1}{Z}\mathbb{E}_{b(s_{t-1})}\Bigg[\sum_{s_t} P(s_t|s_{t-1},a_{t-1};\theta_{12}) \nabla_{\theta_{11}} P(o_t|s_t;\theta_{11}) \\
&\quad + \sum_{s_t} P(o_t|s_t;\theta_{11}) \nabla_{\theta_{12}} P(s_t|s_{t-1},a_{t-1};\theta_{12})\Bigg]
\end{aligned} \tag{3.20}$$

The equality can generally be achieved without requiring the gradients w.r.t. the observation and transition distributions parameters to be all zeros. This shows that, we can hold b_t fixed with different (and potentially infinite) sets of parameters θ. The same analysis can be straightforwardly extended when we also require b_{t+1} to be fixed. This would require computing the gradient of $b(s_{t+1})$ w.r.t. θ and adding the resulting terms to (3.19).

While one may ask whether the set of belief-equivalent parameters will reduce and eventually shrink to a single element if we increase the number of time steps for which we require the belief sequence to be invariant? I will challenge this view using the observation from the classification literature that a discriminative classifier can be parameterized by a potentially infinite set of generative models generating the same predictions [141]. Under this view, our argument for the extension

of the belief unidentifiability result to the multi-step setting can be justified by treating the entire belief sequence as a single classifier prediction. In this context, a classifier can make the correct classification for the wrong reason, i.e., using the wrong generative model.

Combining this result with the policy equivalence relationship in (3.16), we can conclude that joint belief-desire inference with partially observable models is unidentifiable since there exists alternative model parameters that render the belief trajectory the same, and the value function and policy can be held fixed by modifying the reward function using expected changes in the value function (weighted by the beliefs). By similarity to the fully observable case, the unidentifiability result also holds in general in the belief inference case where the reward function is provided.

The analysis of both fully and partially observed MDPs show that the full BTOM inference is unidentifiable as a result of an under-determined system, which leads to the same likelihood for an infinite set of parameter configurations. A natural way to overcome this degeneracy under the Bayesian framework is to impose an informative prior on the parameters. I propose a method to learn such a prior from data in the next section.

3.5 Reconciling Subjective and Objective Models Using Informed Priors

In this section, I seek to alleviate the unidentifiability of BTOM by revisiting its definition in (3.2). (3.2) posits that while making inference about the target agent, the observer assumes agent parameters θ to be independent of the environment parameters ϕ. However, is the independence assumption reasonable? For example, if the target agent was trained in a similar environment, its parameters likely correlate with the current environment. Similar questions have been raised in the context of semi-supervised learning, where learning data-dependent joint priors (a form of empirical Bayes estimation [142]) were proposed to break the independence assumption and make use of unlabeled data [143, 144, 145, 146]. I briefly review these prior-engineering approaches below and propose a method to relate the *objective* and *subjective* environment models.

3.5.1 Bayesian Prior Engineering in Semi-Supervised Learning

In supervised learning, we observe a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ of observations x_i and labels y_i. The goal is to predict the unknown labels of new observations. The dataset is typically modeled as samples from the following joint distribution:

$$P(x, y|\theta, \phi) = P(y|x; \theta)P(x|\phi) \tag{3.21}$$

where $P(y|x; \theta)$ is the classifier we wish to obtain. The parameters are assumed to be independent $P(\theta, \phi) = P(\theta)P(\phi)$.

Given this model structure, the posterior over the classifier parameters is:

$$P(\theta|\mathcal{D}) = \frac{\prod_{i=1}^n P(y_i|x_i; \theta)P(\theta)}{\int_\theta \prod_{i=1}^n P(y_i|x_i; \theta)P(\theta)} \tag{3.22}$$

which is dependent on the data distribution $P(x)$ but independent of the model distribution $P(x|\phi)$.

In semi-supervised learning, we are provided with an additional unlabeled dataset $\tilde{\mathcal{D}} = \{\tilde{x}_j\}_{j=1}^m$. If we use the model defined above and assume the unlabeled dataset is generated from the same distribution, then the unlabeled dataset does not provide any information about the classifier parameters θ. The independence assumption was first questioned by Seeger [143]. In supervised learning, the labels y are usually attributes of the observations x — it is unlikely that θ and ϕ are independent. Thus, the majority of approaches reviewed here propose to model the joint distribution $P(\theta, \phi)$.

The most straightforward method is to assume θ and ϕ are mirror image of each other: $P(\theta, \phi) = \delta(\theta - \phi)P(\phi)$, if $P(y|x; \theta = \phi)$ is modeled as the posterior of a generative model $P(x|\phi)$:

$$P(y|x; \phi) = \frac{P(x|y; \phi)P(y|\phi)}{\sum_{y'} P(x|y; \phi)P(y|\phi)} \tag{3.23}$$

The unlabeled dataset is incorporated in the posterior as:

$$P(\phi|\mathcal{D},\tilde{\mathcal{D}}) = \frac{\prod_{i=1}^{n} P(y_i|x_i;\phi)P(x_i;\phi)\prod_{j=1}^{m} P(\tilde{x}_j|\phi)P(\phi)}{\int_\phi \prod_{i=1}^{n} P(y_i|x_i;\phi)P(x_i;\phi)\prod_{j=1}^{m} P(\tilde{x}_j|\phi)P(\phi)} \tag{3.24}$$

where $P(\tilde{x}|\phi) = \sum_{\tilde{y}} P(\tilde{x},\tilde{y}|\phi)$ is the marginal likelihood of the unlabeled data point.

However, the mirror-image assumption might be too restrictive or even disadvantageous in cases where the generative model $P(x|\phi)$ is misspecified (e.g., the naive Bayes model) or cannot be estimated accurately from a small dataset [147]. To relax the mirror-image assumption, Bishop and Lasserre [146] proposed a joint prior which is proportional to the $l2$ distance between the two sets of parameters:

$$P(\theta,\phi) \propto P(\theta)P(\phi)\frac{1}{\sigma}\exp\left(\frac{1}{2\sigma^2}||\theta-\phi||^2\right) \tag{3.25}$$

where a hyperparameter σ controls the strength of the prior belief of parameter similarity.

Still, the above method requires generative modeling of the data distribution $P(x|\phi)$, which can be difficult and excessive for the downstream classification task. We often wish to directly learn the classifier $P(y|x)$ while still making use of the unlabeled dataset $\tilde{D}$. Grandvalet and Bengio proposed to encode in the prior the assumption that the unlabeled data points tend to have unambiguous labels, motivated by the observation that "the (asymptotic) information content of unlabeled examples decreases as classes overlap" [144, 148]. They proposed to encode this prior as the exponential of the negative entropy of the classifier on unlabeled examples:

$$P(\theta,\phi) \propto \exp\left(-\lambda\mathcal{H}(\tilde{y}|\tilde{x})\right) \tag{3.26}$$

where $\mathcal{H}(\tilde{y}|\tilde{x}) = -\mathbb{E}_{\tilde{x}\sim P(\cdot|\phi),\tilde{y}\sim P(\cdot|\tilde{x};\theta)}[P(\tilde{y}|\tilde{x};\theta)]$ and λ is a hyper parameter controlling the strength of the prior belief of unlabeled class entropy. The expectation over $P(\tilde{x}|\phi)$ may be alternatively estimated from the unlabeled dataset $\tilde{\mathcal{D}}$ and thus avoid generative modeling.

While the above methods were proposed for semi-supervised learning, the important insight

is that the joint prior is more realistic and can be formulated in a dataset-dependent fashion. I introduce a joint prior for the BTOM problem in the next section.

3.5.2 Joint Priors For Environment and Agent Inference

I have shown that the assumption of agent parameters θ being independent of the environment parameters ϕ is conceptually inadequate and can lead to degenerate solutions. A reasonable solution is to require the agent parameters θ to be similar to the environment parameters ϕ. We can formulate this assumption using the KL divergence between the subjective and objective environment models. We consider choices of reverse and forward KL divergence below.

Reverse KL divergence prior. Using the reverse KL divergence, the prior can be formulated as:

$$\begin{aligned} P(\theta, \phi) &\propto P(\theta)P(\phi)\exp\left(\lambda R(\theta_1, \phi)\right) \\ R(\theta_1, \phi) &= -\sum_{t=1}^{T} D_{KL}\left[P(o_t|h_{t-1}, a_{t-1}; \theta_1)||P(o_t|h_{t-1}, a_{t-1}; \phi)\right] \end{aligned} \tag{3.27}$$

where λ is a hyperparameter controlling the strength of such belief. Given the mode-seeking and zero-avoiding property of the reverse KL divergence [76], this prior has the interpretation of constraining the subjective model to generate transitions where the probability under the objective model is non-zero, resonating with the proposal in [109]. The BTOM Bayesian network with joint environment-agent prior is shown in Fig. 3.3 with an additional edge between ϕ and θ compared to Fig. 3.1.

Using this prior, the joint agent-environment BTOM problem can be written as:

$$\begin{aligned} P(\theta, \phi|o_{1:T}, a_{1:T}) &= \frac{P(o_{1:T}|h_{1:T-1}, a_{T-1}; \phi)P(a_{1:T}|h_{1:T}; \theta)P(\theta, \phi)}{\int_{\theta,\phi} P(o_{1:T}|h_{1:T-1}, a_{T-1}; \phi)P(a_{1:T}|h_{1:T}; \theta)P(\theta, \phi)} \\ &= \frac{P(o_{1:T}|h_{1:T-1}, a_{T-1}; \phi)P(a_{1:T}|h_{1:T}; \theta)P(\theta)P(\phi)\exp\left(\lambda R(\theta_1, \phi)\right)}{\int_{\theta,\phi} P(o_{1:T}|h_{1:T-1}, a_{T-1}; \phi)P(a_{1:T}|h_{1:T}; \theta)P(\theta)P(\phi)\exp\left(\lambda R(\theta_1, \phi)\right)} \end{aligned} \tag{3.28}$$

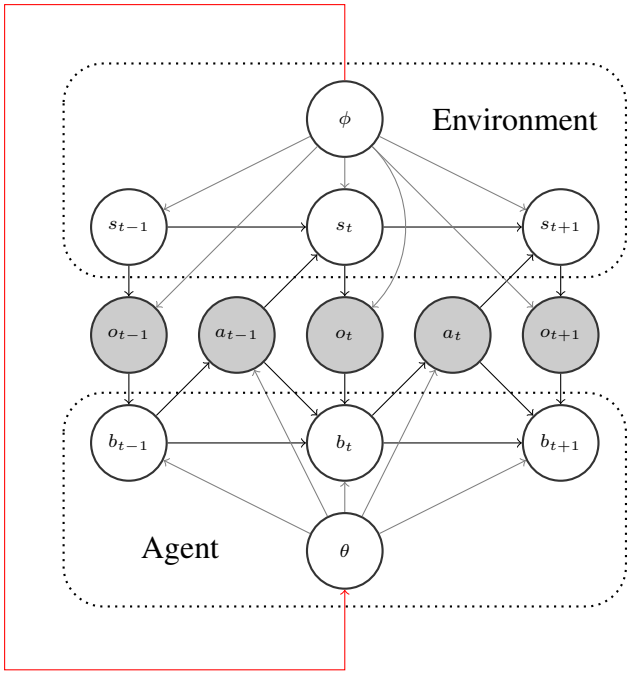


Figure 3.3: Bayesian network of joint environment-agent BTOM. In contrast with Fig. 3.1, an additional edge between ϕ and θ encodes the joint environment-agent dependency.

with the joint MAP estimate equal to:

$$\begin{aligned}
\{\theta, \phi\}^{\text{MAP}} &= \arg\max_{\theta,\phi} \sum_{t=1}^{T} \Big\{ \log P(a_t|h_t;\theta) + \log P(o_t|h_{t-1}, a_{t-1};\phi) \\
&\quad - \lambda D_{KL}\left[P(o_t|h_{t-1}, a_{t-1};\theta_1)||P(o_t|h_{t-1}, a_{t-1};\phi)\right] \Big\} + \log P(\theta) + \log P(\phi) \\
&= \arg\max_{\theta,\phi} \sum_{t=1}^{T} \Big\{ \log P(a_t|h_t;\theta) + \log P(o_t|h_{t-1}, a_{t-1};\phi) \Big\} + \log P(\theta) + \log P(\phi) \\
&\quad + \lambda \sum_{t=1}^{T} \Big\{ \mathbb{E}_{P(o_t|h_{t-1},a_{t-1};\theta_1)}\left[\log P(o_t|h_{t-1}, a_{t-1};\phi)\right] + \mathcal{H}[P(o_t|h_{t-1}, a_{t-1};\theta_1)] \Big\}
\end{aligned} \tag{3.29}$$

The last line shows that this prior favors θ with high objective likelihood and high entropy. At the same time, the objective model is encouraged to have high likelihood under samples from the subjective model. This requires the observer to maintain two copies of the environment model, which may be beneficial when the observer is itself an autonomous agent with its own separate

goals such that an accurate model of the environment is needed. However, this may be excessive if the observer is a passive reasoner, e.g., when we are interested in only target parameters in a psychology experiment.

Forward KL divergence prior. Using the forward KL divergence, we have in the prior:

$$R(\theta_1, \phi) = -\sum_{t=1}^{T} D_{KL}\left[P(o_t|h_{t-1}, a_{t-1}; \phi)||P(o_t|h_{t-1}, a_{t-1}; \theta_1)\right] \tag{3.30}$$

The joint MAP estimate can be written as:

$$\begin{aligned}\{\theta, \phi\}^{\text{MAP}} = \arg\max_{\theta,\phi} \sum_{t=1}^{T} \Big\{ \log P(a_t|h_t; \theta) + \log P(o_t|h_{t-1}, a_{t-1}; \phi) \Big\} + \log P(\theta) + \log P(\phi) \\ + \lambda \sum_{t=1}^{T} \Big\{ \mathbb{E}_{P(o_t|h_{t-1}, a_{t-1}; \phi)} \left[\log P(o_t|h_{t-1}, a_{t-1}; \theta_1)\right] + \mathcal{H}[P(o_t|h_{t-1}, a_{t-1}; \phi)] \Big\}\end{aligned} \tag{3.31}$$

Under the assumption that the ϕ is faithful to the actual environment such that the log likelihood of the subjective model expected under the objective model is equal to that expected under the actual environment, we can write the MAP estimate of θ as:

$$\theta^{\text{MAP}} = \arg\max_{\theta} \sum_{t=1}^{T} \Big\{ \log P(a_t|h_t; \theta) + \lambda \log P(o_t|h_{t-1}, a_{t-1}; \theta_1) \Big\} + \log P(\theta) \tag{3.32}$$

This is equivalent to having a prior:

$$P(\theta) \propto P(\theta) \exp\left(\lambda \log P(o_{1:T}|h_{1:T-1}, a_{T-1}; \theta)\right) \tag{3.33}$$

which removes the requirement of having two copies of the environment model.

It is important to distinguish (3.32) from the two-stage decoupled inference framework reviewed in Section 3.3.2, where an environment model $P(o_{1:T}|h_{1:T-1}, a_{1:T-1}; \theta_1 = \phi)$ is first estimated and subsequently held fixed while making inference about agent desire θ_2. In this method, the environment parameters θ_1 depend on agent actions only via causal conditioning (i.e., Pearl's

do-calculus [106]) and are thus *not* affected by agent decisions. In contrast, in (3.32) the environment parameters θ_1 depend on both agent decisions and the actual environment observations such that agent decision can inform its estimation.

3.6 Summary

In this chapter, I introduced Theory of Mind inference as a framework for behavior understanding, continuing the quest for understanding the coupled roles of perception and action in autonomous agents. I then reviewed relevant literature in desire, belief, and joint TOM inference and applications, illustrating the use cases and advantages of TOM over alternative human understanding frameworks. Although empirically observed in the literature, the unidentifiability of TOM was largely unexplored. I provided an analysis of TOM identifiability with MDP and POMDP models and showed that they suffer from being under-determined systems, leading to degeneracy in the likelihood function. To overcome this degeneracy, I proposed a family of informed, joint agent-environment priors which can be estimated from data. These priors are more consistent with human TOM inference than uninformative priors [14] and unify the heuristic regularization approaches in current belief inference algorithms [109]. Subsequent sections illustrate insights about human driving behavior made possible by the proposed priors.

4. MODELING DRIVER RESPONSES TO AUTOMATION FAILURES WITH ACTIVE INFERENCE*

4.1 Summary

The goal of this chapter is to investigate how active inference and Bayesian theory of mind can be applied to model actual human behavior. Specifically, I study how active inference can be used to understand driver emergency braking decision process during a simulated laboratory driving task with automated driver assistance system. A central challenge in this task is in understanding the source of heterogeneity in driver behavior. A model that does not take into account heterogeneity will generate poor predictions of driver behavior. Using a combined active inference-BTOM-expectation maximization approach, I show that the heterogeneity in driver behavior can be understood as varied beliefs about the road condition and driver assistance system to automatically recover from crash risk. This provides a general methodology for understanding naturalistic human behavior.

4.2 Introduction

Automated vehicle (AV) technologies promise to substantially reduce the 1.35 million annual worldwide roadway fatalities [149], yet preliminary deployments of AVs have had mixed results. While there is evidence that AVs improve safety–measured by crashes per mile [150, 151]–there is corresponding counterevidence in the form of fatal crashes [152] and difficulties during interactions with other vehicles [8, 9]. These crashes have a diverse set of causes, but most involve a mismatch between driver expectations and automation capabilities [153]. The effects of these mismatches are most insidious after automation failures where drivers need to re-engage with the driving task and avoid a crash [154]. Reducing such crashes requires developing AVs that are designed within human capabilities and expectations. Integrating models of human perception and

action in the design process will facilitate such designs [155].

Driver process models (DPMs) are a promising method to integrate human perception and action with the design process. DPMs are a class of models that can specify momentary control actions given prior input from the driver, system, and surrounding driving environment [156]. DPMs can be used to simulate driver behavior and driver-system interactions in counterfactual situations. The outcomes of these simulations can, in turn, be used to assess safety outcomes and calibrate system parameters within driver limits [157, 158]. Given that the role of DPMs is to emulate driver behavior, it is critical that the predicted model behavior and decision processes align with actual observed driving behavior and decision processes. In the context of interactions with AVs, models need to emulate driver perception of the environment and beliefs about their responsibilities (i.e., their expectations) [154].

Prior models of driver behavior following AV failures have mostly used visual looming of obstacles in the forward roadway to model drivers' perception of the environment [159]. Visual looming is derived from the optical angle of an object in the forward view subtended on the driver's retina, and is defined by the ratio of the change in optical angle and the optical angle itself. It is well established in the literature that visual looming is the central source of perceptual evidence that drivers use to initiate braking behavior in rear-end braking emergencies [160, 154, 161, 162]. Prior modeling efforts have focused on predicting driver braking reaction time (BRT) and behavior from accumulated visual looming evidence over time [163, 164, 165]. Recent work has extended these findings to show that after prolonged use of automation, driver braking behavior is more accurately predicted by accumulated errors in expected and observed visual looming [166, 167]. Beyond the evidence accumulation framework, Pekkanen et al. [168] developed a model that integrates perceived visual looming, with driver state estimation, environmental parameters, and actions. While these models have shown considerable promise for emulating human braking behavior, they are limited in the sophistication of their representation of cognitive states and in their applicability to scenarios involving multiple decisions (e.g., car following). These limitation are especially relevant for AVs given that internal states such as trust [169, 170, 153] and situation awareness

[171, 172, 173] substantially affect driver responses to automation failures.

One method of addressing this gap is to extend models of human perception and action that are grounded in cognitive theory and neurological behavior to the driving domain (e.g., [174]). The most prominent of these approaches is active inference [22]. The central tenets of active inference are: 1) human decision-makers embody an internal model of the environment which they use to minimize an information theoretic measure of surprise called free energy and 2) all relevant parameters of the model (including perception, action, and others) are optimized in a Bayesian fashion.

The ideas of active inference have been proposed as a general theory of driver behavior [175], but they have not been extended to quantitatively model driving behavior. The goal of this article is to report on the development of a novel model of driver behavior that integrates active inference and visual looming. We accomplish this goal through proposing and parameterizing an active inference model of driver braking reactions in rear-end braking scenarios, then by demonstrating how the model parameters can be mapped to known psychological constructs, and illustrating counterfactual predictions made possible by the model.

4.3 Active Inference in a Partially Observable Markovian Environment

Active inference has been used to model human decision-making under uncertainty in partially observable environments. This partial observability follows from the observation that the brain does not have direct access to the true environmental state but must infer it from noisy sensory signals [18]. In this section, we provide a brief overview of active inference as a model of human decision-making and introduce relevant constructs including states, actions, preferences, free energy, variational inference, and Partially Observable Markov Decision Processes (POMDP).

4.3.1 Active Inference: The Observable States Case

Given a finite set of observable states $\mathcal{S}$, individual preferences are modeled by a probability distribution $P(s) \in (0, 1)$ with $\sum_{s \in \mathcal{S}} P(s) = 1$, such that high probabilities correspond to states with high expected visitation. Preferences can be modeled by means of rewards $r(s)$ so that the

relative log-likelihood of state s over s' is proportional to the difference of rewards:

$$\log \frac{P(s)}{P(s')} \propto r(s) - r(s') \tag{4.1}$$

In a static decision-making environment, the agent (i.e., driver) is assumed to have a (potentially inadequate) internal model for the consequences of its actions in the form of a *predictive* distribution $Q(s|a)$ over states conditioned upon decision (or action) $a \in \mathcal{A}$. The Kullback–Leibler divergence (also called relative entropy) between distributions $Q(s|a)$ and $P(s)$ is defined as:

$$\mathcal{G}(a) = D_{KL}(Q(s|a) \| P(s)) := \sum\nolimits_{s \in \mathcal{S}} Q(s|a) \log \frac{Q(s|a)}{P(s)}$$

This measure is often referred to as the expected free energy (EFE) [29]. It measures the difference between *preferred* states likelihood and *predicted* states likelihood and can be re-written as:

$$D_{KL}(Q(s|a) \| P(s)) = \mathbf{E}_{s \sim Q(\cdot|a)}[-\log P(s)] - \mathbf{H}(Q(\cdot|a))$$

where the first term is expected "surprise" (i.e., disagreement between the desired vs. predicted states under action a) and the second term $\mathbf{H}(Q(\cdot|a))$ is the entropy of the *predictive* distribution $Q(s|a)$. Let π denote a probability distribution over $\mathcal{A}$. In active inference, meaningful behavior is modeled by the relative log-likelihood of selecting action a over a' as follows:

$$\log \frac{\pi(a)}{\pi(a')} \propto -\gamma(\mathcal{G}(a) - \mathcal{G}(a'))$$

with $\gamma > 0$ a precision parameter controlling the concentration of the action distribution π.

In a dynamic setting, we can denote by $s_{1:T} = (s_1, s_2, \ldots, s_T)$ a sequence of observable states with $s_t \in S$, $t \in \{1, \ldots, T\}$. Also, we can denote by $a_{1:T} = (a_1, a_2, \ldots, a_T)$ a sequence of actions $a_t \in \mathcal{A}$, $t \in \{1, \ldots, T\}$. With a Markovian *predictive* distribution $Q(s_{t+1}|s_t, a_t)$ we can extend

the definitions above (with s_1 given) as follows:

$$Q(\, s_{1:T} |\, a_{1:T}) = \prod_{t=1}^{T} Q(\, s_{t+1} |\, s_t, a_t) \tag{4.2}$$

$$P(s_{1:T}) = \prod_{t=1}^{T} P(s_t) \tag{4.3}$$

The EFE can be extended as follows:

$$\mathcal{G}(a_{1:T}) = D_{KL}(Q(s_{1:T}|a_{1:T})\|\, P(s_{1:T})) \tag{4.4}$$

As before, meaningful behavior is cast as the log-likelihood, $\log \pi(\, a_{1:T} |\, s_1)$, of selecting action a proportional to $-\mathcal{G}(a_{1:T})$, i.e.,

$$\pi(\, a_{1:T} |\, s_1) \propto \exp(-\gamma \mathcal{G}(a_{1:T})) \tag{4.5}$$

4.3.2 Variational Inference

In the case wherein the state is not observable but the agent is able to record an observation $o \in \mathcal{O}$ with joint distribution $P(o, s) = P(o|s)P(s)$, $s \in \mathcal{S}$ where $P(s)$ is the a priori distribution and $P(o|s)$ is the observation probability. The a posteriori distribution is

$$P(s|o) = \frac{P(o|s)P(s)}{\sum_{s \in \mathcal{S}} P(o|s)P(s)}$$

In active inference modeling, the computation of the a posteriori distribution is generally assumed to be intractable. Variational inference is an alternative approach where an $approximation$ to the a posteriori distribution is obtained by solving the following optimization problem [66]:

$$Q^*(s|o) = \arg \min_{Q(\cdot) \in \mathcal{Q}} \sum\nolimits_{s \in \mathcal{S}} Q(s|o) \log(\frac{Q(s|o)}{P(s|o)}) \tag{4.6}$$

where $\mathcal{Q}$ is a class of conditional probability distributions parameterized by "free variational" parameters.

4.3.3 Application to POMDP

At a given time t, the agent interacts with the environment by performing an action $a_t \in \mathcal{A}$ with effects a transition from the current state s_t to the next state s_{t+1} with the dynamics governed by a probability distribution $P(s_{t+1}|s_t, a_t)$. The state $s_t \in \mathcal{S}$ is not directly observable; yet the agent registers an observable signal $o_t \in \mathcal{O}$ generated through probability $P(o_t|s_t)$, which is used to infer the underlying state using variational inference, i.e., (4.6). Fig. 4.1 illustrates this process.

As before, active inference models the agent's decision-making by minimizing EFE. Since the environment is partially observable, the agent evaluates EFE expected under the future sequences of observations $P(o_{1:T}|s_{1:T})$ as follows: [68]:

$$\begin{aligned} \mathcal{G}(a_{1:T}) &= \mathbf{E}_{P(o_{1:T}|s_{1:T})}[D_{KL}(Q(s_{1:T}|a_{1:T})|\, P(o_{1:T}, s_{1:T}))] \\ &= D_{KL}(Q(s_{1:T}|a_{1:T})\|\, P(s_{1:T})) \\ &+ \mathbf{E}_{Q(s_{1:T}|a_{1:T})}[\mathbf{H}(P(o_{1:T}|s_{1:T}))] \end{aligned} \tag{4.7}$$

where the first term in the second line is the same as (4.4) and the second term is the entropy of observations expected under the predictive distribution $Q(s_{1:T}|a_{1:T})$. Action selection again follows (4.5).

4.4 Active Inference Braking Model

In this section, we illustrate how the active inference framework presented in the previous section can be applied to model a driver's emergency braking behavior. We achieve this by formulating the mental model of the driver, specifying its action selection mechanism, and grounding the observation process in visual looming observations.

4.4.1 Active Inference Braking Model Formulation

Following the active inference framework, we assume the driver consistently updates beliefs about the state of the environment and the actions to pursue in order to maintain consistency with the expected state distribution $P(s)$, up to a planning horizon H unknown to the researchers. We focus on the driver's decision-making process for the braking reaction event, which begins at the moment the lead vehicle begins to decelerate. Thus, the driver only considers two actions: waiting (0) and braking (1).

We assume the driver mentally represents the environment with $K > 1$ states, each associated with a visual looming observation distribution $P(o|s)$. This follows from the observation that drivers' braking decision-making is guided by visual looming [162]. The states can thus be interpreted as either more or less urgent based on their associated looming expectations being higher or lower. The impact of the braking and waiting actions on the states as perceived by the driver is described by the driver's internal model of state transitions in the environment $P(s_{t+1}|s_t, a_t)$. The driver uses the state transition model for two purposes: 1) to form beliefs about the state of the environment (i.e., whether a crash is imminent) through sequences of past visual looming observations as in (4.6), and 2) to mentally simulate and predict sequences of future states should a sequence of actions be taken as in (4.2). The parameters of the observation and state transition distributions (illustrated by the squares in Fig. 4.1) are internal to the driver and unknown to the researchers. However, the relationships between these parameters and the actions taken by the driver, which are observable to the researchers, are described by the expected free energy functional.

Consistent with [29], we calculate the driver's mental simulation of state sequences when implementing different actions and the resulting EFE using dynamic programming. Assuming at any given time t, the driver believes they will choose actions that minimize the EFE exponentially more likely, the EFE (4.7) of choosing action a_t while the inferred state estimate is s_t can be computed

using the following recursive equations [29]:

$$\begin{aligned}\mathcal{G}(s_t, a_t) = & D_{KL}(Q(s_{t+1}|s_t, a_t) \| P(s_{t+1})) \\ & + \mathbf{E}_{Q(s_{t+1}|s_t,a_t)}[\mathbf{H}(P(o_{t+1}|s_{t+1}))] \\ & + \mathbf{E}_{Q(s_{t+1},a_{t+1}|s_t,a_t)}[\mathcal{G}(s_{t+1}, a_{t+1})]\end{aligned} \tag{4.8}$$

The last term, corresponding to the expectation of the EFE at the subsequent time step, is zero at the last time step $t = H$. This assumption reduces searching among all action sequences to only searching among the most likely action sequences (i.e., that achieve lower EFE), by pruning away action sequences that are unlikely (i.e., high EFE) a priori.

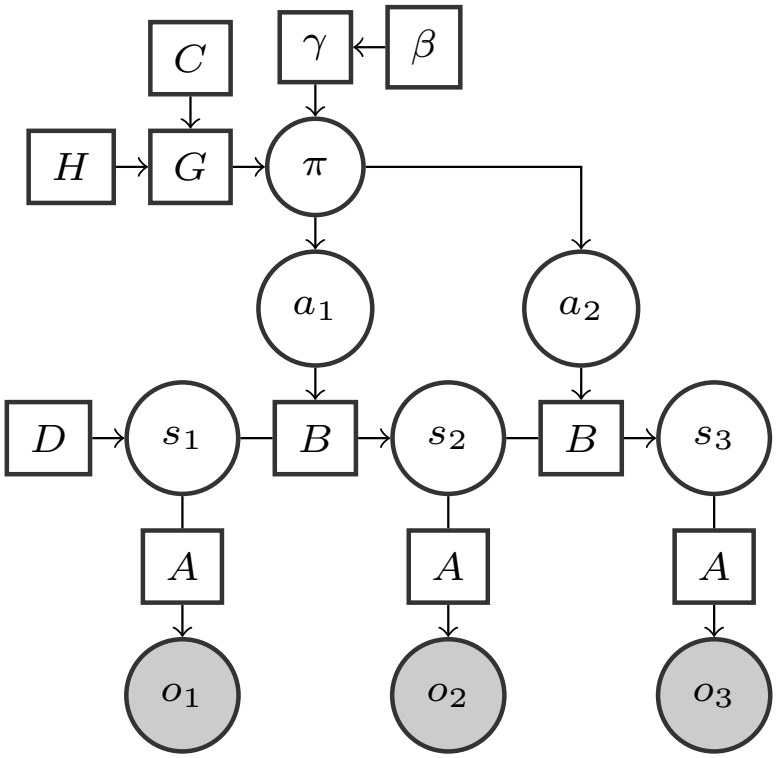


Figure 4.1: Factor graph illustration of an active inference agent in a POMDP environment. The circles represent random variables and squares represent parameters internal to the agent. Observable variables in the environment are colored in gray. The figure shows three time steps of interactions with the environment through observations o and actions a, which the agent uses to form beliefs about the environment $Q(s_{1:T}|a_{1:T})$ and actions to pursue $\pi(a_{1:T})$.

Following [67], we assume the driver also updates belief about precision γ at each time step in addition to states and actions. It can be shown that if we equip the driver with a gamma prior over precision with shape parameter α (usually set to 1) and adjustable rate parameter β, the rate parameter update is proportional to the difference between the prior and posterior evaluations of the EFE [67]: $\Delta\beta_t = -(\pi_t - \pi_{t-1})\mathcal{G}_t$ (for detailed derivation see Section B.2). Thus, precision reflects the driver's sensitivity to the accuracy of their own evaluation of EFE. We include this parameter because prior studies have shown that active inference models with a dynamic precision inference mechanism better resemble human choice behavior compared to models without precision [176, 177]. In addition, the research has shown that observed changes in precision estimate correlate with neurological activities (e.g., dopamine)[176].

4.4.2 Mapping Model Components to Constructs

A distinct advantage of this active inference braking model is that the internal model components align with known psychological constructs that are relevant to driving behavior and transitions of control. The expected state distribution encodes the driver's prior expectations of the *long-term* future states, where high probability corresponds to what the driver thinks is likely to happen when the driver-environment system is in a steady-state. The driver in turn adapts their behavior to maintain consistency with the expected steady states (e.g., maintaining task-difficulty homeostasis)[178]. The mean of the observation model conditioned on each state $P(o|s)$ encodes the looming value the driver expects to see under that state. The variances of the observation model relate to attention mechanisms [179]. Specifically, overt attention corresponds to having an observation model with low variance. This leads to a higher divergence between the observation distribution conditioned on different states, resulting in a higher rate of perceptual evidence accumulation and a higher ability to distinguish among alternative states [180, 181]. The state transition model encodes the driver's understanding of the environment dynamics and prediction of the *immediate* next state based on the current state. To predict over multiple time steps, the driver cascades single-time step predictions from the state transition model. Finally, the precision parameter γ is related to the flexibility and automaticity of action selection [175]. This follows from the

mechanism by which precision modulates action selection: high precision increases the probability of selecting actions evaluated to have low EFE even though the evaluation may be inaccurate due to few observations or an inadequate model, while low precision discourages this behavior and encourages sampling more observations before making a decision. Thus, drivers with high precision behave more autonomously and are less affected by sensory observations, while drivers with low precision behave with more variance.

4.5 Methods

We analyzed the active inference braking model by fitting it to driving data from a driving simulation study and analyzing the fitted parameters. Following the model parameterization process, we used a factor analysis to identify factors that concisely explain variation in the model parameters and further evaluated the model's ability to make counterfactual predictions on unseen scenarios. This section provides details of these procedures.

4.5.1 Dataset

The dataset used for parameter estimation was obtained from a driving simulator study of driver responses to automation failures while driving in a platoon [182]. The study was approved by the Texas A&M Institutional Review Board (IRB number: IRB2018-1362D) and complied with the American Psychological Association's code of ethics. The study had a 2x2x2 factorial design, where the presence of an alert (alerted vs. silent) was varied between subjects, and the takeover scenario (unexpected braking vs. obstacle reveal) and scenario criticality (critical vs. non-critical) were within subjects. After practice sessions to familiarize participants with the simulator and its automation capabilities, participants completed four experiment drives corresponding to each pairing of post-failure driving environment and scenario criticality. The order of the drives was counterbalanced across participants. For the purpose of the current analysis, only the unexpected braking scenario was included as initial observations suggested that this scenario mostly produced braking responses, whereas drivers typically responded with steering in the obstacle reveal condition.

In each experiment drive, the participant drove on a highway with a speed limit of $105kph$ ($65mph$) and they were instructed to keep their hands on the steering wheel. The drive started with the participant's vehicle on the side of the road near a highway entrance ramp where participants were instructed to enter the highway behind a lead vehicle and engage the vehicle's automation. The participants then drove for approximately 10 minutes with the automation engaged, although they were permitted to disengage the automation at any point by pressing the brake pedal or a button on the center console. At approximately 3 minutes into the drive, the lead vehicle braked and the participant's vehicle responded with equivalent braking. After approximately 7 minutes, the lead vehicle braked a second time, the participant's vehicle failed to respond and the automation disengaged. The lead vehicle deceleration rates in the critical and non-critical scenarios were $5m/s^2$ and $2m/s^2$, respectively. In the alerted condition, this disengagement coincided with an auditory and visual alert and in the silent condition the participant received no indication of automation disengagement.

Sixty-four participants (32 males, 32 females, mean age 41.44 years (SD = 15.14)) completed the study, generating 256 drives. Each drive contained $10Hz$ position, velocity, acceleration, and brake pedal position data. The remainder of this analysis focuses on the 128 drives from the unexpected braking scenarios, specifically the second braking event in which the automation failed to respond. Analysis of the other responses is reserved for future work. In addition to the removal of the obstacle reveal scenario, 20 drives in the braking condition where drivers steered without braking were also excluded. The final dataset included in this analysis consists of 108 drives with 51 drives in the critical scenarios and 57 drives in the non-critical scenarios. Of these drives, 3 resulted in crashes—2 from the silent failure and critical condition and 1 from the alerted and critical condition. A Bayesian regression analysis of braking reaction times found that there were substantial differences between the critical and non-critical conditions in braking response time (mean increase of 0.76s in the non-critical condition compared to the critical condition), however, there was not a substantial difference between the silent and alerted conditions (mean increase of 0.16s in the silent failure condition). This finding was attributed primarily to the fact that drivers

were instructed to keep their hands on the wheel and did not engage in secondary tasks while the automation was engaged [182]. In addition, no substantial effects of driver demographics (i.e., age and gender) or condition order were observed. Additional details on the study, including descriptive statistics, can be found in [182, 183].

4.5.2 Data Pre-processing

Given that the focus of the current analysis is on driver responses to automation failures, we subset the data from each drive to include only the second braking event (at 7 minutes into the drive) and specifically the time period from the initial braking of the lead vehicle until the observed braking reaction time of each drive–brake pedal depression of 1% or greater. We calculated the looming value at each time step following the method in [162]. For each drive, the sequence of actions implemented by the driver is a sequence of zeros, corresponding to wait, except for the last action being 1, corresponding to brake, at the observed BRT. We added 2 additional time steps (0.2 seconds) after the observed BRT with braking actions to each sequence in order to avoid the model overfitting to a single braking action. Thus, for each drive, the processed dataset consists of a sequence of continuous looming values and a sequence of binary actions.

4.5.3 Parameter Estimation

Since our dataset is constrained in size, we adopted a Bayesian approach and estimated *distributions* over the drivers' (i.e., the participants') internal parameters under the active inference model. This approach allows us to quantify uncertainty over the estimated parameters and avoid parameter overfitting.

We parameterized the models with the complete set of active inference parameters for a fixed number of environment states K described in Sec. 4.4.1. These parameters include the looming observation distributions $P(o|s)$ with parameters $A = \{A_i\}_{i=1}^{K}$, state transition dynamics $P(s'|s,a)$ with parameters $B = \{B_{ij}^{a}\}, i,j \in \{1,...,K\}, a \in \{1,...,|\mathcal{A}|\}$, expected state distribution $P(s)$ with parameters $C = \{C_i\}_{i=1}^{K}$, initial state belief $P(s_0)$ with parameters $D = \{D_i\}_{i=1}^{K}$, initial precision rate β, and planning horizon H. Since looming values are non-negative, we modeled

the looming observation distributions using log-normal distributions with location and scale parameters: $A_i = \{\mu_i, \sigma_i\}$. We parameterized our belief over the drivers' planning horizons using a Poisson distribution with rate τ. We found using the number of states $K = 2$ was sufficient to recover observed braking behavior. This resulted in 12 effective parameters, i.e., $K - 1$ parameters for each probability vector. The relationships between these parameters and other variables in the driver-environment system are shown in Fig. 4.1.

The model parameter estimation was conducted with the Empirical Bayes method [142], a hierarchical Bayesian model which uses the empirical dataset to generate an informed prior distribution over the parameters and in turn constrain the parameter space. We estimated a separate posterior distribution for each drive in order to understand the behavioral variations in the dataset. Grouping all model parameters into a single vector $\theta = \{A, B, C, D, \beta, \tau\}$, the suitability of the parameters for an individual drive is determined by the likelihood of action sequence $a_{1:T}$ taken by the driver given the observed looming sequence $o_{1:T}$ of the respective drive and the parameters θ:

$$\begin{aligned} P(a_{1:T}|o_{1:T}, \theta) &= \prod_{t=1}^{T} P(a_t|o_{1:t}, \theta) \\ &= \prod_{t=1}^{T} P(a_t|Q(s_t), \theta) P(Q(s_t)|Q(s_{t-1}), o_t, a_{t-1}) \end{aligned} \tag{4.9}$$

where $P(a_t|Q(s_t), \theta) \propto \exp(-\gamma \mathcal{G}(s_t, a_t))$. We approximated the calculation of $\mathcal{G}(s_t, a_t)$ in (4.8) using the QMDP method [42].

We performed the Empirical Bayes estimation using a Variational Expectation Maximization algorithm [66]. The objective of the algorithm is to maximize the log-marginal likelihood:

$$\mathcal{L}(a_{1:T}|o_{1:T}) = \log \int P(\theta) \prod_{t=1}^{T} P(a_t|o_{1:t}, \theta) d\theta \tag{4.10}$$

where $P(\theta)$ is the Empirical Bayes prior over the parameters. We overcame the intractable integral using a variational posterior distirbution $Q(\theta)$ for each drive, which gives rise to the following

lower bound on the log-marginal likelihood (derived in Appendix Sec. B):

$$\mathcal{L} = \mathbb{E}_{Q(\theta)}[\sum_{t=1}^{T} \log P(a_t|o_{1:t}, \theta)] - D_{KL}[Q(\theta)||P(\theta)] \tag{4.11}$$

We used a multivariate normal distribution with full covariance matrix for the prior distribution in order to capture the relationships between the (log-transformed if applicable) model parameters. This is equivalent to using normal distributions for μ, log-normal distributions for σ, β, and τ, and logistic-normal distributions for B, C, and D. We used multivariate normal distributions with diagonal covariance for the variational posteriors $Q(\theta)$ to simplify optimization.

Since our dataset is constrained in the diversity of observations, i.e., all looming sequences increased monotonically as a result of the deterministic braking scenarios, maximizing the action likelihood alone as in (4.10) can lead to unrealistic solutions. Hence, we further constrained the model with an observation likelihood regularizer:

$$\mathcal{L}(o_{1:T}) = \sum_{t=1}^{T} \log \sum_{s_t} Q(s_t) P(o_t|s_t) \tag{4.12}$$

in order to avoid unrealistic parameters with low observation likelihood and avoid overfitting to driver action sequences. We added the observation regularizer to the action likelihood objective with an adjustable penalty coefficient of 0.2.

The involvement of latent variables (i.e., the active inference model parameters θ) in the estimation procedure is prone to local optima. Thus, we performed the optimization algorithm with multiple random initialization of parameters. Among these random initializations, we selected the parameters which achieved the highest log-likelihood and the lowest Kolmogorov–Smirnov (KS) distance between the predictive and empirical BRT distributions in the prior and posterior predictive checking processes described in Sec. 4.5.5. We provide further details of our optimization procedure in the Appendix B.

4.5.4 Model Parameter Analysis

Following model parameter estimation, we performed a factor analysis to further understand the relationships between model parameters and the resulting behavior, and mapped correlations between parameter changes and BRT to psychological constructs relevant to transitions of control (e.g., trust). We employed the approach described in [184], which assumes that each observed data instance is generated by first drawing a latent vector z from a normal distribution $\mathcal{N}(z|0, I)$ with zero means and identity covariance matrix I, and then sampling the observed data x from a linearly transformed normal distribution $\mathcal{N}(x|Wz + \mu, \Sigma)$ with diagonal covariance matrix Σ. The factor analysis procedure consists of recovering the parameters W, μ, and Σ, and interpreting the latent factors based on the recovered parameters. In the current context, we expect the latent variable z to encode intrinsic properties of the drivers, which we can elicit by interpreting the loading matrix W.

We fit the factor model to the braking model parameters estimated using the procedure described in Sec. 4.5.3, where the parameters estimated for each drive correspond to a data instance. To improve the quality of the recovered factors, we applied log-transformation to all non-negative parameters and standardized all parameters [184]. The optimal number of factors was identified through an iterative comparison of factor models including between 1 and 10 factors. The models were evaluated with data log-likelihood and Bayesian Information Criteria (BIC). The optimal number of factors was selected as the point of maximum curvature identified by the Kneedle algorithm [185].

4.5.5 Model Validation

We validated the Empirical Bayes model and the factor model by drawing samples of active inference parameters from the models and simulating agents with the sampled parameters in randomly generated novel scenarios. We first generated scenarios similar to the experiment by fitting a t-distribution to the initial speeds of participant vehicles and a half-normal distribution to the initial distances to the lead vehicle, drawing samples from each distribution, and calculating the

resulting looming sequences assuming the participant never braked and the lead vehicle followed the experiment scenarios, including the critical and non-critical scenarios. We used these scenarios to perform prior and posterior predictive checking by simulating active inference agents with parameter sets drawn from the prior and posterior distributions. We recorded the simulated BRTs as the time since the beginning of the simulation and the first braking action executed by each active inference agent.

4.5.6 Counterfactual Simulation

We examined the fitted model's generalization capability by performing a counterfactual simulation of a rear-end emergency scenario where the ego vehicle responded with braking after a fixed time delay. This scenario emulates a case where a secondary safety system, e.g., automated emergency braking (AEB), activates after an automation failure. In this case, the time delay between activation of the secondary safety system may be an important design variable. The counterfactual simulation had the same initial parameters as the experiment. At the start of the scenario, the lead vehicle was $30m$ (i.e., the average distance from the experiment) ahead of the ego vehicle and both vehicles were traveling at $105kph$ ($65mph$). After the first time step, the lead vehicle initiated a deceleration of $2m/s^2$ and the ego vehicle continued at $105kph$ ($65mph$). The initial time-to-collision was $5s$. After a variable time delay, the AEB system in the ego vehicle responded with a deceleration of $5m/s^2$. Five AEB time delays were considered: 0.5, 1.5, 2.5, 3.5, and 4.5 seconds. At each time delay, we sampled 3,000 sets of parameters from the factor model and simulated the active inference agents for $5s$ with these parameters on the looming values generated by the counterfactual scenarios to identify the predicted agent's time-to-decision (TTD). Each agent executed one drive in each delay scenario.

4.6 Results and Discussion

4.6.1 Model Fitting and Validation

4.6.1.1 Model Validation

The average expected log-likelihood of driver action sequences was -0.4, or equivalently 0.98 per time step (0.1 second). Fig. 4.2 shows the prior and posterior predictive distributions of BRT compared with the empirical distribution. The KS distances between the prior predictive and the empirical distribution and the posterior predictive and the empirical distribution were 0.284 and 0.162, respectively. The close alignment between the cumulative densities shows that the fitted model captured the observed behavior well in all drives and a uni-modal prior distribution was appropriate.

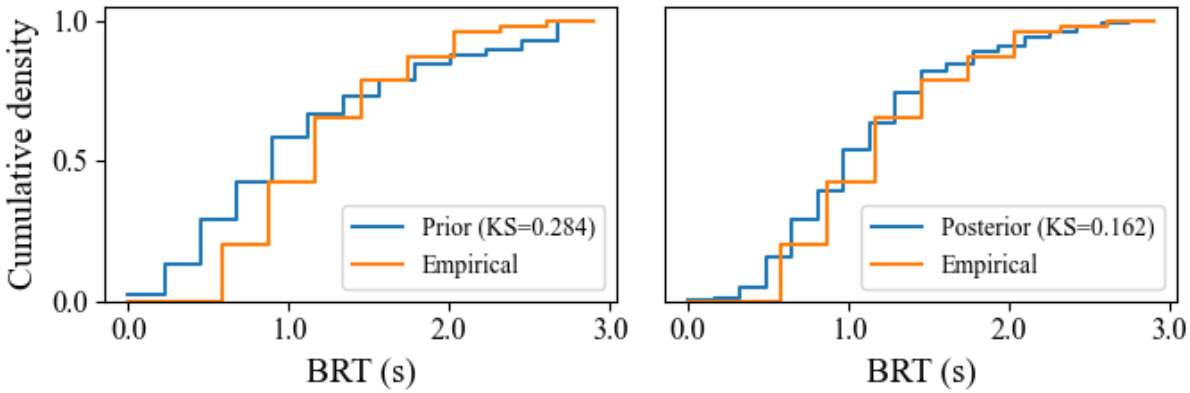


Figure 4.2: Prior (left) and posterior (right) cumulative predictive distributions of braking reaction times compared with the empirical braking reaction times. The KS distances between the prior/posterior distribution and the empirical distribution are shown in the legends.

4.6.1.2 Fitted Parameters

Prior to subsequent analysis, we compared the fitted parameters across participant age and gender groups and found no significant differences. Fig. 4.3 shows violin plots of the posterior distributions of driver parameters aggregated over all drives in each scenario. Each chart in the

figure corresponds to an active inference parameter, and the widths of the violins correspond to the density of the distributions. The majority of the distributions were uni-modal, except parameters $A_{\mu 0}$, $A_{\mu 1}$, $A_{\sigma 0}$, and $A_{\sigma 1}$ for the non-critical scenarios were multi-modal. Most parameters were observably skewed, shown by the asymmetric shapes of the violins. These two properties of the posterior distributions suggest variations between the posterior distributions across different drives. The modes of the posterior distributions of parameters $A_{\mu 0}$ and $A_{\mu 1}$ for all scenarios, corresponding to the modes of the looming observation distributions for states 0 and 1, were 0.1 and 0.02, respectively. Thus, states 0 and 1 were associated with high and low expected looming observations, corresponding to the urgent and non-urgent state, respectively. The modes of the posterior distributions for parameters $A_{\sigma 0}$ and $A_{\sigma 1}$ for all scenarios, corresponding to the looming observation distributions A_{σ} for the urgent and non-urgent states, were 0.02 and 0.08. This shows that the drivers had more precise looming expectations for the urgent state than the non-urgent state.

The posteriors of all state recurrence rate parameters B were distributed between 0.8 and 1, with B^0_{00} distributed most densely between 0.96 and 0.99, B^0_{11} between 0.925 and 0.975, B^1_{00} between 0.8 and 0.9, and B^1_{11} between 0.8 and 1. The high values of all state recurrence rates show that the drivers expected both states to independently reoccur with high probability, effectively considering the environment to be close to static such that it is either urgent or non-urgent but does not transition between the two. This is consistent with the experiment scenario, where the lead vehicle maintained a constant deceleration rate for 5 seconds and caused looming values to increase monotonically. The lowered state recurrence rates when taking the braking action show that the drivers expected to influence the environment state through braking.

The posteriors of the initial belief parameter D_0 were most densely distributed between 0 and 0.4. This shows that the drivers recognized the scenario was not urgent before the lead vehicle initiated braking. The expected state distribution parameter C_0 was most densely distributed between 0.6 and 0.9. The posterior of planning horizon τ was most densely distributed between 2 and 4 seconds and the posterior of precision γ was most densely distributed between 5 and 10. This shows

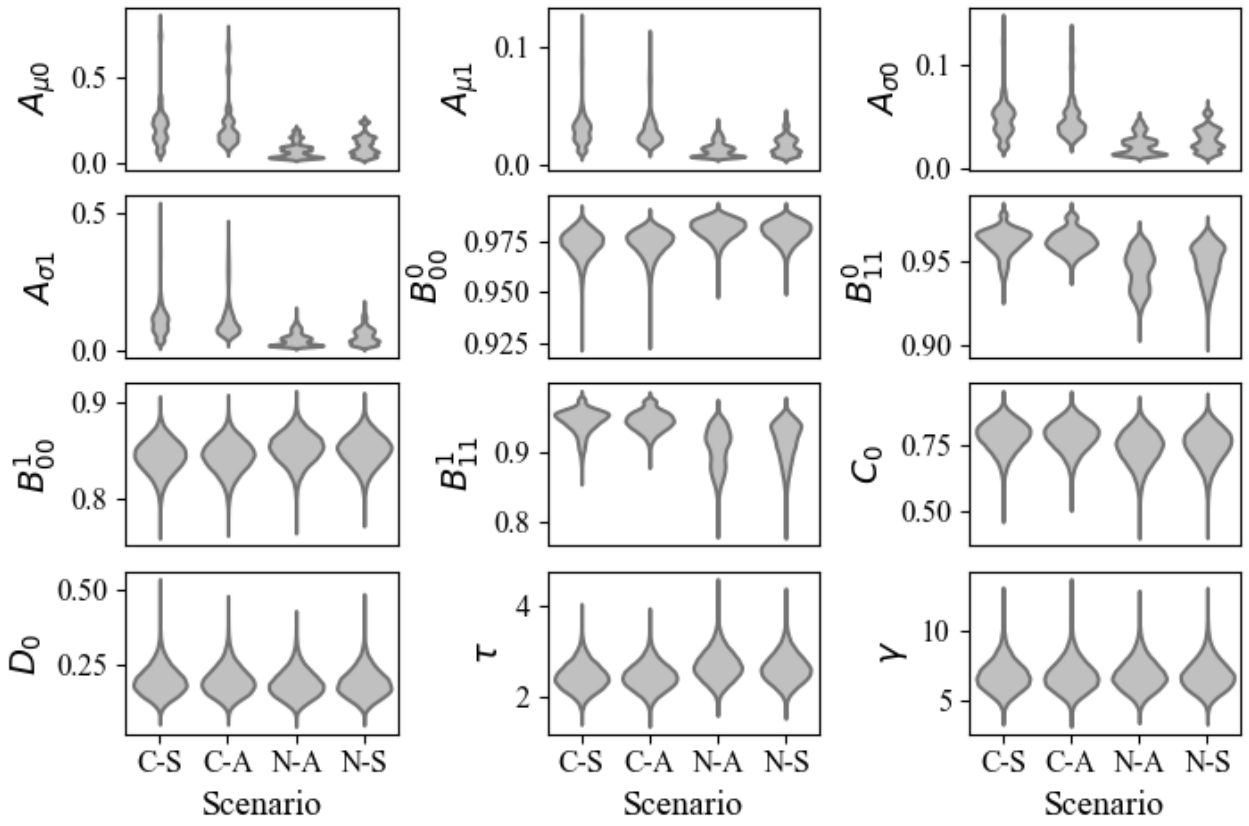


Figure 4.3: Posterior distributions of active inference parameters aggregated over all drives. Each chart corresponds to an active inference parameter. Each violin in a chart corresponds to an experiment scenario, with shorthand *C = Critical, N = Non-critical, A = Alerted, S = Silent*. The grey violins represent the density of the posterior samples, where wider regions correspond to higher densities. Parameters with subscripts 0 and 1 are associated with the urgent and non-urgent states, respectively. Parameters with superscripts 0 and 1 are associated with the waiting and braking actions.

that while the drivers planned their actions for multiple time steps, the braking decision process overall was still relatively reactive.

4.6.1.3 Between-Trial Comparison

Fig. 4.3 suggests there were noticeable parameter differences between the critical and non-critical scenarios but not between different alert conditions. Specifically, the differences were seen in the A, B, C, and τ parameter distributions, corresponding to the drivers' looming observation, state transition, expected state distribution, and planning horizon parameters. These variations

were expected as the drivers showed different behavior likely due to the difference in their beliefs about the environment. However, there were no noticeable differences in the initial belief and precision parameters across different experiment conditions. Post-hoc analysis indicated that the variations were likely not due to overfitting to driver actions or observations, as both conditions achieved similar average action and observation likelihood. Thus, the variations in parameters were most likely due to differences in the duration of the decision-making process. We further explored this insight in subsequent sections by extrapolating the parameter variations with a factor analysis and observing the resulting behavioral variations in simulations.

4.6.2 Model Analysis

In the model fitting and validation step, we found there were noticeable differences between the posterior parameter distributions of different drives. Thus, we used the maximum a posteriori (MAP) parameters to represent the posterior distributions of each drive and performed all subsequent analysis on the MAP parameters.

4.6.2.1 Factor Analysis

Fig. 4.4 shows the log-likelihood and BIC results for the iterative factor analysis. Higher log-likelihood values and lower BIC values indicate a better model fit. Each point in the chart represents an iteration of the analysis between 1 and 10 factors. Based on these results, 4 factors were identified as the optimal value by the Kneedle algorithm as it achieves the highest curvature of the BIC curve.

The factor loading matrix (top) and the explained variance (bottom) for the 4-factor model are shown in Fig. 4.5. The top heatmap in the figure shows the loading on each parameter associated with each factor, and the bottom heatmap shows the fraction of variance explained in each parameter by the four factors combined. The cells of the heatmap are color-coded with red indicating a positive relationship and blue indicating a negative relationship between the parameter values and the factor. The opacity of the cells reflects the amount of loading on each factor in the top plot and the amount of variance explained in the bottom plot with darker cells representing higher loading

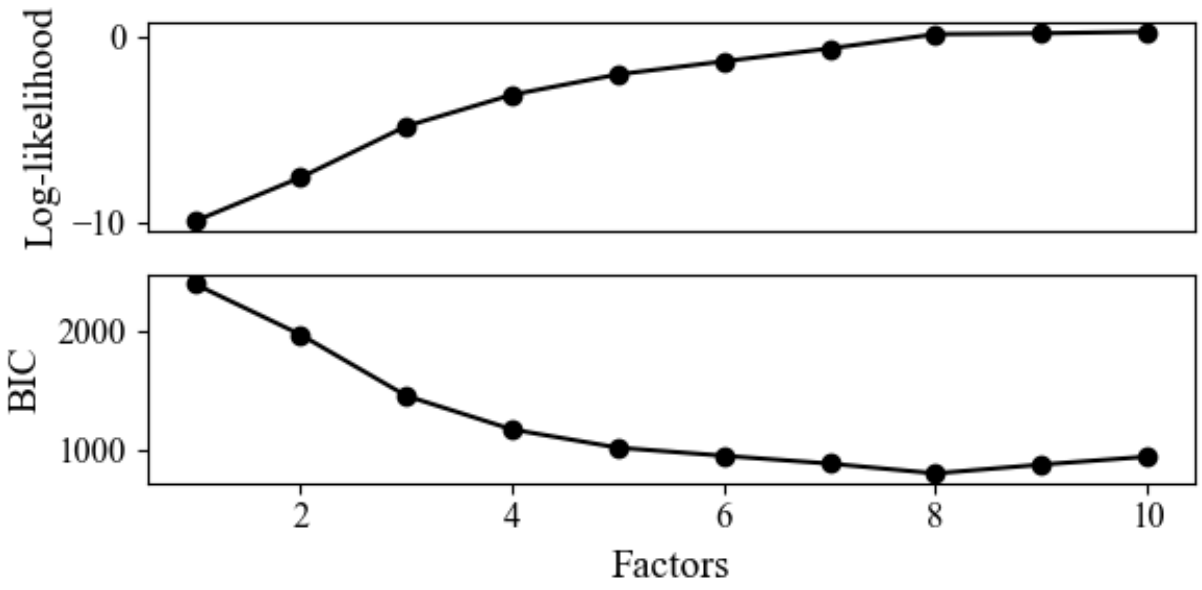


Figure 4.4: The log-likelihood and BIC values by the number of factors. Four (4) factors were selected as the optimal number of factors.

and more variance explained, respectively. Specifically, factors 1-3 were primarily loaded on B^1_{00}, D_0 and C_0, respectively, with loading values of 0.88, -0.98, and -0.84. Thus, factors 1-3 captured simple variations in individual parameters corresponding to the drivers' belief of state transition when braking in the urgent state, initial belief, and expected state distribution. In contrast, factor 4 has high loading values on parameters $A_{\mu 0}$, $A_{\mu 1}$, $A_{\sigma 1}$, B^0_{00} B^0_{11}, B^1_{11}, and τ. Thus, factor 4 captured more complex relationships between a separate set of parameters from factor 1-3, corresponding to the drivers' expectations of observed looming values, belief of state transitions when not braking, belief of state transition when braking in the non-urgent state, and the planning horizon.

The fraction of variance explained for the majority of the parameters were close to 1, except that $A_{\sigma 1}$, τ, and γ had explained variances of 0.36, 0.69, and 0.01, respectively. This shows that the factor model captured the variance in the drivers' expectations of states and observations well. The very low variance explained by γ shows that precision had no obvious correlation with other parameters and the drivers' braking behavior.

The mapping from parameter variations to behavioral variations can be visualized by plotting

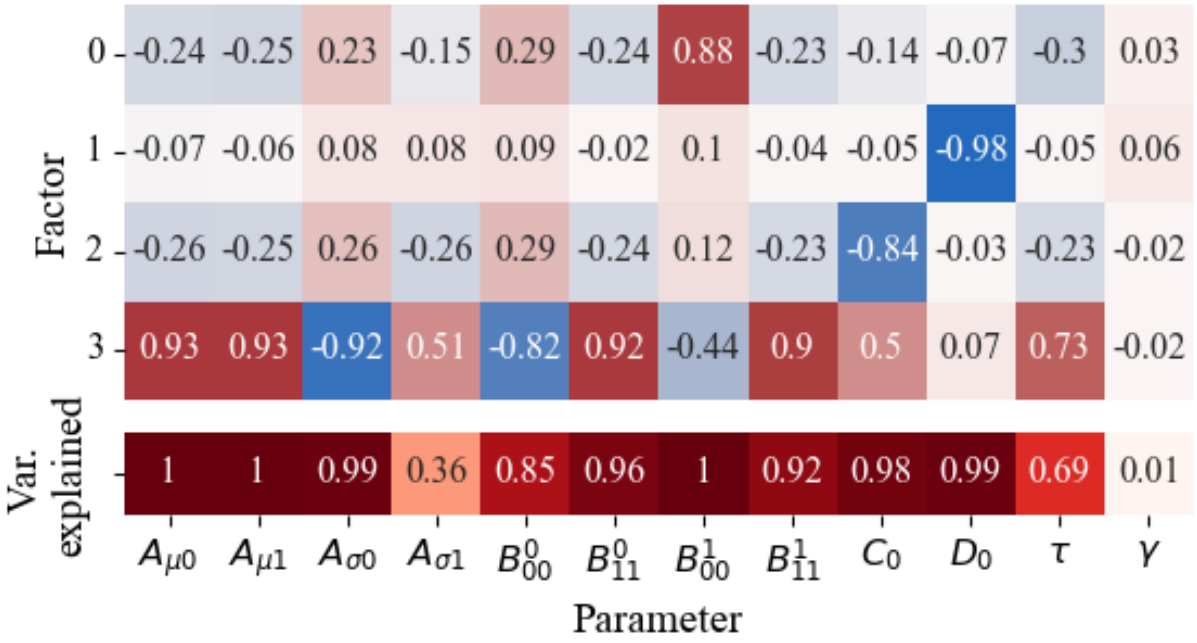


Figure 4.5: Factor analysis results. Row 1-4 shows the factor loading matrix. Row 5 shows variance explained by the factor model. Each column corresponds to an active inference model parameter. Parameters with subscripts 0 and 1 are associated with the urgent and non-urgent states, respectively. Parameters with superscripts 0 and 1 are associated with the waiting and braking actions. The value in each cell corresponds to the estimated factor model parameter value.

the recovered factor values for each drive against the observed BRTs. This relationship, along with the factor distributions across these drives, is shown in Fig. 4.6. The distributions of all factors were uni-modal and centered at 0 with empirical ranges of 5, and factor 4 was more densely distributed on the left of 0. There were no obvious relationships between the values of factor 1-3 and the empirical BRTs, whereas the value of factor 4 was positively correlated with BRTs. This suggests that the factor model has captured variations in driver behavior in addition to driver parameters.

Fig. 4.7 further highlights the relationship between the factors and BRT. The figure shows the pairwise influence between factor 4 and factors 1-3 on BRT (plotted with a color scale). Lower predicted BRT values are shown in purple while higher predicted BRT values are yellow. Each plot shows BRT for a uniform random sample of 3,000 active inference parameters sets across the latent factor space. Each point in the plots represents one sampled parameter set. This sampling

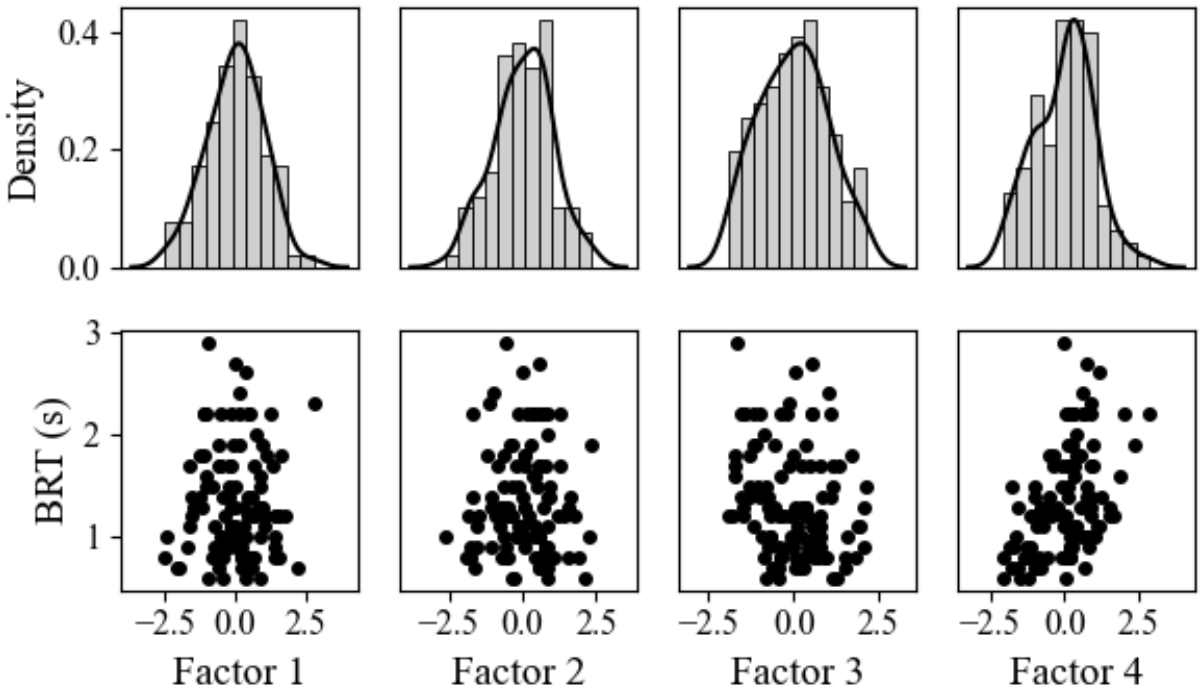


Figure 4.6: Distributions of the factors across the dataset (top) and the relationship between factors and observed BRT (bottom).

method was used to make the relationships between factors clearer compared to the sparse sample of drivers from the experiment. The figure shows that independent of factors 1-3, lower values of factor 4 led to shorter BRT and higher values of factor 4 led to longer BRT. However, the plot also highlights that factors 1 and 3 interact with factor 4, given that the ratio of purple to yellow points in the bottom left (factor 1) and bottom right (factor 3) changes over the values of factor 1 and factor 3, respectively. Specifically, higher values of factor 1 and 3 are associated with shorter BRT. The figure also suggests that of all the factors, factor 2 had the least influence on BRT in the observed data.

4.6.2.2 Factor Interpretation

The factor analysis results show that the estimated parameters of the drivers can be summarized with 4 orthogonal factors. Given the close mapping between the active inference parameters and psychological constructs (Sec. 4.4.2), we can expect the factors to represent semantics of behavior

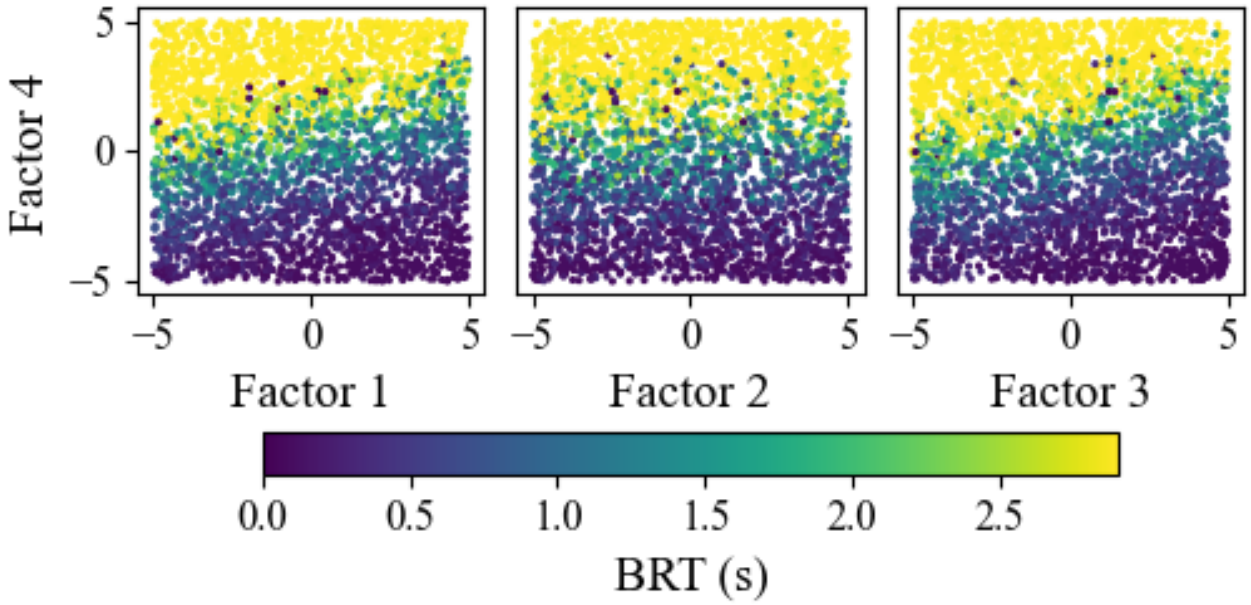


Figure 4.7: Visualization of interactions between factors and predicted BRT. The points represent 3,000 simulated parameter sets uniformly sampled from the factor model. Each point is color-coded by its predicted BRT in a non-critical scenario, with yellow representing high BRT and purple representing low BRT.

as variations in and interactions between the individual parameters. We consider a loading value greater than 0.5 to be substantial and representative of the factors and we focus on the interpretations of these values.

The sparsity of loading values on factor 1-3 suggests that these factors represent simple semantics. Specifically, with a single high loading value of 0.88 on B^1_{00}, factor 1 represents drivers' expected recurrence of the urgent state when braking. With a loading value of -0.98 on D_0, factor 2 represents drivers' initial belief. With a loading value of -0.84 on C_0, factor 3 represents drivers' expected state distribution. These factors correspond well with the variations in BRT shown in Fig. 4.7, where higher values of factors 1 and 3 were associated with shorter BRT while factor 2 had no observable influence on BRT. This is because higher value of B^1_{00} in factor 1 and lower value of C_0 in factor 3 lead to the drivers' belief of lower divergence between the preferred and predicted future states (4.8) when taking the braking action. On the other hand, prior belief does not have an obvious effect on BRT because the looming observation distributions of the urgent and non-urgent

states are different enough that drivers can accurately estimate the state after a few observations.

In contrast with factor 1-3, factor 4 represents more complex semantics as it has a higher number of large loading values compared with factor 1-3, with high positive loading values on $A_{\mu 0}$, $A_{\mu 1}$, B^0_{11}, B^1_{11}, and τ, and high negative loading values on $A_{\sigma 0}$ and B^0_{00}. Drivers with positive values of factor 4 are associated with increased expected looming observation values, increased expected recurrence of the non-urgent state while taking both actions, and longer planning horizon. They are also associated with decreased looming observation variance for the urgent state, corresponding to more precise looming expectations, and decreased expected recurrence of the urgent state while not braking. This depicts a driver who expects the non-urgent state to occur more often and urgent state to disappear more often, behaves less reactively, and expects to see higher looming in both states. As such, factor 4 corresponds well with the concept of “Trust", defined as “the attitude that an agent will help achieve an individual’s goals in a situation characterized by uncertainty and vulnerability" [186], as the altered expectations of state transitions and observations can be attributed to the driver’s belief in the automation’s ability to intervene in a near crash scenario. On the other hand, factor 4 can also be related to the concept of “Situation Awareness", defined as “the perception of the elements in the environment [...], the comprehension of their meaning and the projection of their status in the near future" [187], as variations in the expected looming observation and state transition parameters can be interpreted as inaccurate perception and prediction of the environment. Both interpretations correspond well with the observation in Fig. 4.6 and Fig. 4.7 where drivers with high values of factor 4 are associated with slower braking. However, it is important to note that while we connect the modeled factors with human factors concepts, we do not claim that the factors indeed represent any particular concept but merely treat them as latent factors of variations in observed behavior.

4.6.3 Counterfactual Simulation

The results of the counterfactual simulations, described in Sec. 4.5.6, are illustrated in Fig. 4.8. Each violin plot in the figure shows the TTD distribution of the 3,000 simulated parameter sets sampled from the factor model. In each scenario, simulated TTDs were most densely distributed

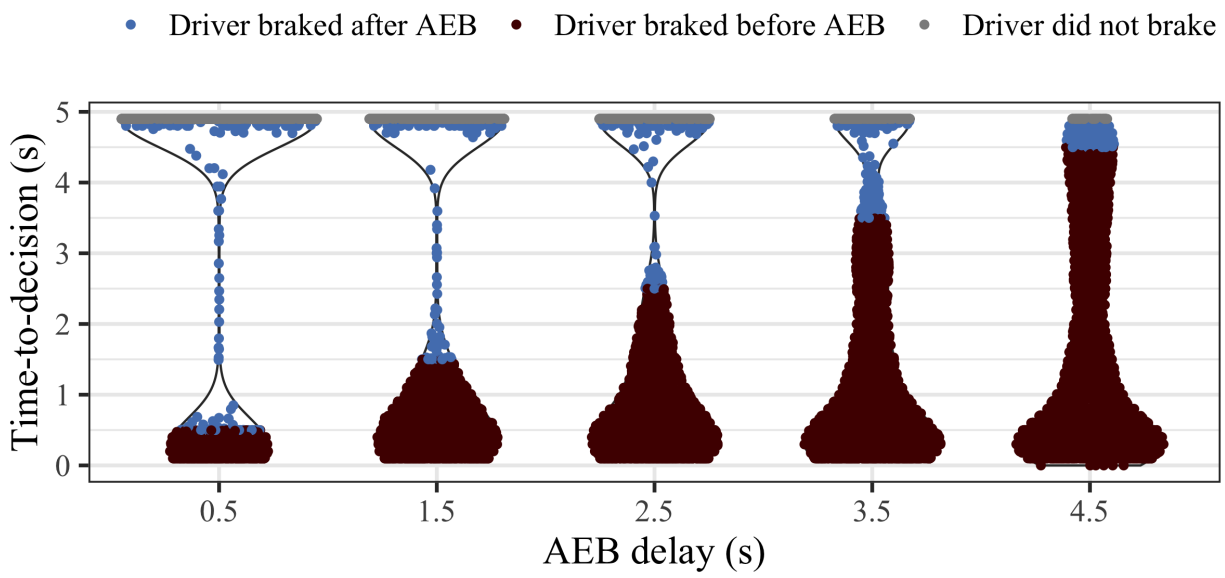


Figure 4.8: Predicted time-to-decisions of 3,000 simulated parameter sets with parameters uniformly sampled from the factor model. Each violin plot shows the TTD distribution of the 3,000 drivers in the corresponding automated emergency braking activation delay scenario with wider regions correspond to higher densities.

at either 0.1 seconds, or 4.9 seconds. However, in high AEB delay scenarios, there were higher densities in the violin plots between 0.5 and 4 seconds. Thus, the figure shows three types of behavior: 1) drivers who braked before the AEB activated (maroon points), 2) drivers who did not respond (i.e, relied on the automation to brake; grey points), and 3) drivers who observed the situation for a period of time before eventually taking over and braking (blue points). While it is difficult to directly compare these results to other studies, it is notable that this pattern of behavior aligns with prior observations of the distribution of driver responses to AV failures [153].

Fig. 4.9 shows the interaction between factor 1, 3, 4, and TTD in the counterfactual scenarios. Each column in Fig. 4.9 corresponds to an AEB delay. The top row shows the interaction between factor 1 and factor 4 , and the bottom row shows the interaction between factor 3 and factor 4. The color of each point represents the predicted TTD of the respective simulated active inference agent parameters, with purple corresponding to immediate braking decision, and yellow corresponding to later braking decision. In all columns, there was a boundary separating the purple region (i.e.,

fast responding drivers) from the rest (i.e., slow responding drivers). This boundary had a positive slope in both rows. This shows that a group of drivers with low values of factor 4 and high values of factor 1 and factor 3 always braked immediately. In the high AEB delay (more than 2.5 seconds) scenarios, the yellow region near the boundary was replaced by blue. This shows that more drivers near the center of the latent factor space started to brake, and this corresponds to the increasing density in the middle sections of the violin plots in Fig. 4.8.

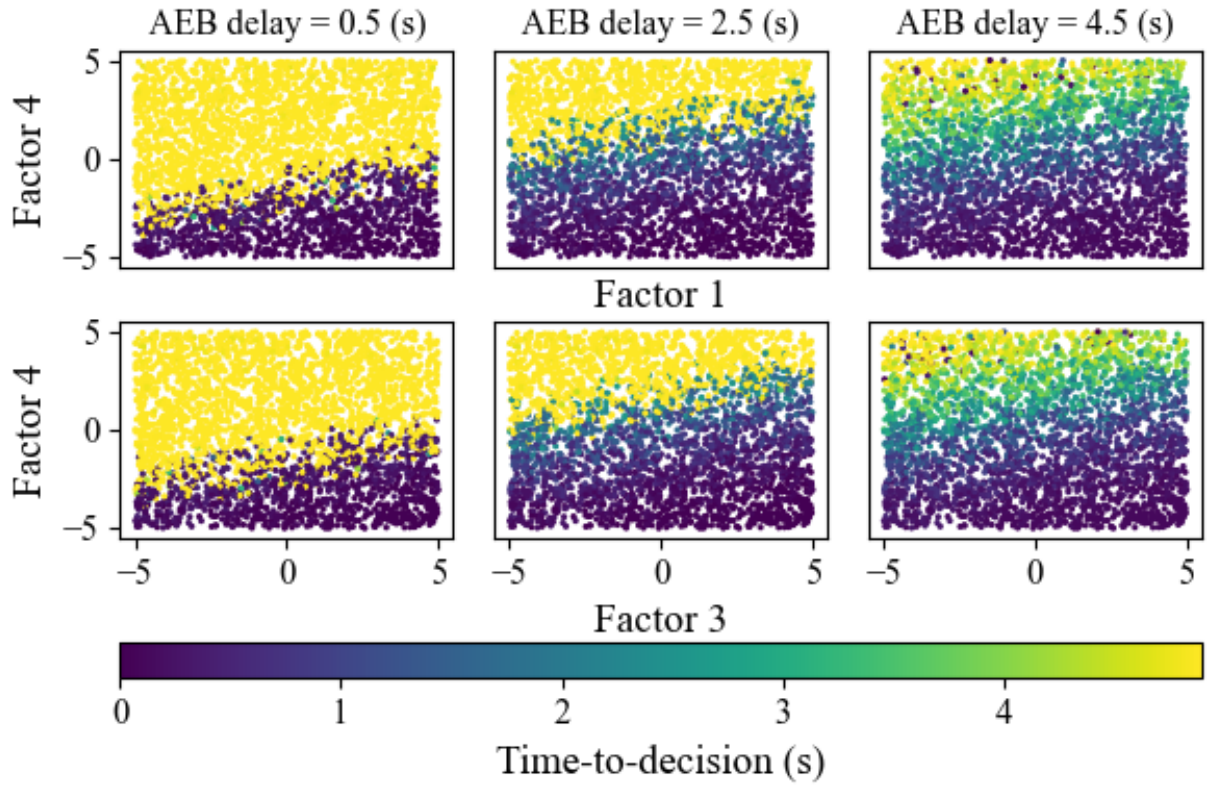


Figure 4.9: Results of the counterfactual simulation. Each column corresponds to an automated emergency braking activation delay. Each point in a subplot corresponds to one of the 3,000 simulated parameter sets uniformly sampled from the factor model. The points are color-coded by predicted time-to-decisions of the simulated parameter sets. Purple corresponds to drivers who braked immediately, and yellow corresponds to drivers who either braked late or did not brake at all. The top and bottom rows show the interaction of factor 1 and factor 3 with factor 4.

Furthermore, as the AEB delay increased, the behavior of drivers at the boundary (i.e., the

region where the purple and yellow meet) for factor 1 and 2 changed in a similar fashion. In both factors, as AEB delay increased from 2.5 to 4.5 seconds, the boundary proceeded upward with similar positive slopes, corresponding to decreased TTDs as factor 1 and 3 vary from negative to positive. This suggests the braking reaction pattern found in a fixed scenario in Fig. 4.7 was invariant under changing scenarios. The similarity between factor 1 and 3 further shows that, while the active inference parameters were different for drivers with different values of factor 1 and 3, the behaviors they generated were similar. This was expected as the state transitions and the expected state distribution play similar roles in the expected free energy functional; changing one while fixing the other should not cause substantial change to the resulting behavior.

4.6.4 General Discussion

In this article, we developed a novel active inference model of driver braking reaction and applied this model to driver braking behavior data following an automation failure. The model not only reproduced the observed behavior of the participants, but also captured their internal decision-making mechanics. A factor analysis showed that the variations in the estimated models can be summarized with a small number of factors and varying these factors led to meaningful change in driver behavior. We further tested the model using counterfactual simulations where the vehicle automation responded to a rear-end braking emergency with different delays and showed that the model can produce expected behavior in new settings.

The counterfactual simulations suggest that factor 4, which can be interpreted as either trust or situation awareness, is the most important factor in determining driver braking response times. This finding is important given that both trust and situational awareness are known to affect driver responses to AV failures [173]. Victor et al. [153] found that several drivers with their hands on the wheel and eyes on the forward roadway still crashed into obstacles in the forward roadway. Further analysis suggested that these crashes could be attributed to high trust measured by perceived capability of the vehicle's automation. This result is consistent with the trust factor identified in the current model. The model suggests that drivers tend to consider high looming as benign when trusting the automation.

The most closely related to our model is the looming evidence accumulation models proposed by Markkula and colleagues [162, 159, 165, 188, 166]. In the looming evidence accumulation models, drivers respond when the accumulated looming evidence, which is the time-integral of noisy looming signals, has exceeded a predefined threshold. Bitzer et al. [180] showed that this evidence accumulation process is equivalent to the active inference belief updating scheme in (4.6) with an identity state transition matrix corresponding to a static rather than a dynamic environment. Action selection in the looming evidence accumulation model follows a threshold policy, where a driver brakes once the belief over the urgent state has exceeded the threshold. This threshold is usually fixed by the modeler when implementing the evidence accumulation model. In our model, the policy threshold is jointly defined by the expected state distribution and the uncertainty encoded in the observation and state transition models, as both are involved in the evaluation of the expected free energy of actions. Thus, our proposed model is a generalization of the evidence accumulation model that allows the state transition and the policy threshold to be estimated from data when manual specification is difficult. Our results show that these generalizations enabled the active inference model to capture more complex cognitive states and dynamics than evidence accumulation models and show how those states and dynamics impact driver reaction times.

Our model also extends previous attempts to implement predictive processing models of driver behavior. In [167], the authors incorporated the idea of predictive processing by positing that drivers react to accumulated looming prediction error rather than looming itself, and looming prediction error is calculated as the difference between the observed looming signal and that predicted with a perfect dynamics model of the vehicle automation. In contrast, our model uses a simpler dynamics model represented by two abstract states. Our model also complements the work of Pekkanen et al. [168] who proposed to model attention as driven by the level of uncertainty over vehicle control states (defined by the expected standard deviation of vehicle acceleration) exceeding a threshold. Although we did not specifically model online adjustment of attention, the variances of observation distributions in our model implicitly captured the influence of attention in the observation variance parameters in factor 4 and subsequently BRTs as shown in the sensitivity

analysis and counterfactual simulations.

Despite our promising findings, this work is limited in the following aspects. First, the model was developed with driving simulator data rather than real-world driving data. While there is an established precedent for relative validity of driving simulation results [189], absolute validity cannot be guaranteed. Second, the dataset used to estimate the model parameters was constrained in size and diversity. The repeated scenarios of the experiment may also have influenced the fitted parameters due to learning effects. Although the fitted model has shown expected behavior in counterfactual simulations, it is unlikely to generalize beyond emergency braking scenarios without training on a larger and more diverse set of data. Finally, while there are associations between the factors we identified in the factor analysis and psychological states, we did not explicitly test these connections. Future work should address these issues by fitting the model to a larger naturalistic dataset, and use experiments to further evaluate the connections between the latent factors, model parameters, and psychological constructs.

4.7 Conclusion

In this work, we developed a new model of driver behavior that leveraged the active inference framework to predict driver braking responses and cognitive dynamics during automation failures and we performed a factor analysis to relate trends in the model parameters to observed behavior. Our analysis and simulations provide novel insight on the behavioral patterns associated with these factors and driver behavior during transition of control. The model offers advantages compared to previous models as it directly measures cognitive dynamics and is more readily scalable to complex driving behaviors (e.g., car following) while maintaining interpretability. Future work should focus on these extensions and validating the findings here with a larger naturalistic driving dataset.

5. SCALING ACTIVE INFERENCE DRIVER MODEL: ADVANTAGES AND APPLICATIONS IN CAR FOLLOWING*

5.1 Summary

The goal of this chapter is to investigate how active inference can be scaled to model human control behavior in more realistic driving scenarios and the advantages of active inference as a modeling paradigm compared to established rule-based and black-box driver behavior models. Specifically, I benchmark active inference against two standard rule-based and data-driven driver models in a highway car following task. Active inference shows competitive performance with the added benefit of superior interpretability provided its modular structure. Crucially, the interpretability enables straightforward inspection and correction of model failures caused by limited data. These results establish active inference as an intermediate driver modeling framework which can incorporate the relative strength of purely rule-based and purely data-driven approaches.

5.2 Introduction

The rapid development of automated and connected vehicle technologies has created a corresponding demand for models of driver behavior that can be used to calibrate design parameters [190, 191], evaluate technologies [192, 157], and refine real-time decision making [193]. To be effective in these tasks, driver models must be flexible, generalizable, and interpretable. Model flexibility is the ability of the model to mimic nuanced social behavior of human drivers [8]. Generalizability is the ability of the model to extend to new environments with minimal modeler intervention. Interpretability refers to both a clear connection between model mechanics and predicted behavior and a grounding in human psychology [194]. These elements facilitate model inspection and diagnostics which are essential for interpretable models [195]. Car following is an important driving sub-task as it represents a large portion of current driving time and crashes involving auto-

*Reprinted, with permission, from Wei, R., McDonald, A. D., Garcia, A., Markkula, G., Engstrom, J., & O'Kelly, M. (2023). An active inference model of car following: Advantages and applications. arXiv preprint arXiv:2303.15201.

mated vehicles [196, 197]. Moreover, it requires a complex expression of social behaviors through physical vehicle positioning [8], e.g., speeding up to prevent a vehicle from merging. Therefore, it is important to develop flexible, generalizable, and interpretable car following models for automated vehicles and future transportation systems.

Existing car following models can be partitioned into rule-based models and data-driven models [161, 154]. Rule-based models generate acceleration behavior based on a function specified by the modeler. Typically, this function is grounded in known observations or driver behavior theory [161]. For example, the Intelligent Driver Model (IDM) predicts driver acceleration based on deviations from a desired speed and distance headway [198]. While rule-based models have a clear connection between model mechanics and predicted behavior, they are limited in their flexibility and generalizability. Because the rules in rule-based models are designed to replicate driving behavior in specific contexts and depict driver characteristics with small parameter sets, they are limited in the behavioral repertoire and in generalizing to scenarios outside of those governed by rules beyond their initial rule set. For example, research has shown that rule-based models designed for car following do not generalize to emergency scenarios and crashes [199]. Despite these limitations, rule-based models are still widely used for automated vehicle analyses [200] and thus offer a valid benchmark for new models.

In contrast to rule-based models, data-driven models learn a function mapping observations or features to acceleration behaviors using an algorithm. Recent works have used neural networks [201], hybrid neural network algorithms with physics constraints [202], reinforcement learning [203], and adversarial imitation learning [204] to model car following behavior. These approaches have shown considerable flexibility in replicating human behavior, however, data-driven models still struggle to reproduce well-known traffic phenomena such as stop-and-go oscillation and their generalizability is constrained by the chosen machine learning technique [201, 205]. Furthermore, the complexity of existing data-driven models prohibits interpretability both in the connection between input and output and in their grounding in human psychology. Despite these shortcomings, data-driven models are more generalizable to complex scenarios which are difficult for manual

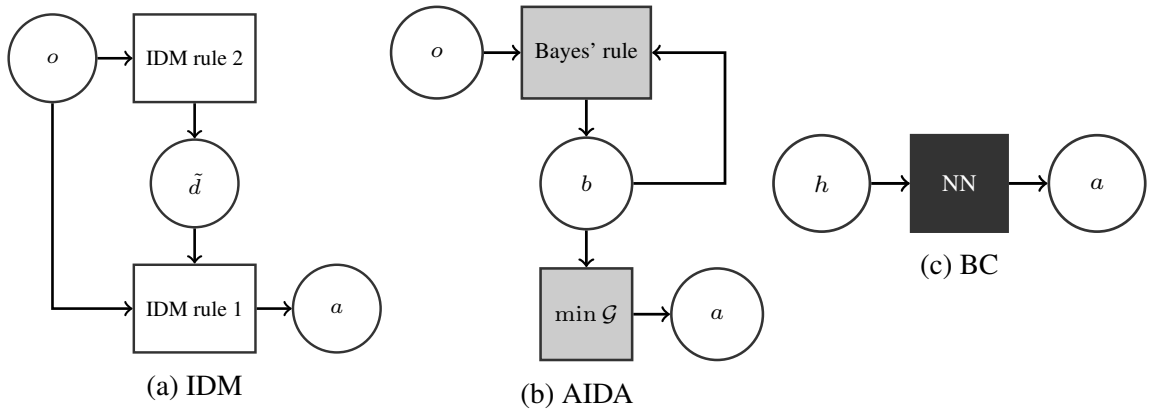
model specification. One important class of data-driven models is Behavior Cloning (BC) known for their simplicity and general effectiveness [206, 207, 208]. Neural network-based BC models have been widely adopted for developing and evaluating automated vehicle algorithms and are a common benchmark for evaluating novel data-driven models [209, 210].

The relative strengths of rule-based and data-driven approaches suggest that there is a role for model structure (to aid in interpretability) — especially structure grounded in psychological theory [194] — and learning from data (to aid in flexibility) in car following model development. The incorporation of these two concepts requires a shift to contemporary theories of human cognition. One relevant theory is active inference [22, 18] — a framework developed from Bayesian principles of cognition [211, 101]. The central ideas of active inference are that 1) humans have internal probabilistic generative models of the environment and that 2) humans leverage their model of the environment to make inferences about action courses that reduce surprise in terms of both distance from their desired states of the environment and uncertainty [22, 18]. Importantly, these principles have been translated into a quantitative framework for modeling human behavior and cognition [18, 135]. The quantitative framework includes an explicit representation of agent belief dynamics to facilitate agent decision making and action selection in response to observed perceptual signals. Due to this structure, the model is fundamentally interpretable (i.e., actions can be traced back to beliefs and observations at a given time). On the other hand, the increased complexity and probabilistic nature of the model compared to rule-based frameworks also increase its flexibility and potentially its generalizability. Recently, the active inference framework has been extended to driving to depict driving behavior during emergency scenarios with some success [15, 212], however, the application to broader scenarios has been limited.

Our goal in this article is to introduce the Active Inference Driving Agent (AIDA), evaluate its flexibility and generalizability relative to rule-based and data-driven benchmarks, and illustrate the interpretability of the model and the resulting insights it provides into car following behavior.

5.3 Materials and Methods

In this section, we introduce a formulation of the benchmarks — IDM and Behavior Cloning — then describe our AIDA formulation. We then describe the dataset used for model fitting and the model comparison approach. To simplify notation, we adopt a unifying view of car following models as longitudinal driving control policies which map input signals observed by drivers to a control signal, i.e., the instantaneous longitudinal acceleration. We denote the driver observations (or features in machine learning terminology) at discrete time step t by o_t and the control signal by a_t. Using this nomenclature, the most generic class of driver control policies can be described as a probabilistic mapping from the entire history of inputs and controls, denoted by $h_t = \{o_{1:t}, a_{1:t-1}\}$ to the next control signal, i.e., $\pi(a_t|h_t)$. However, the control policy may only depend on the most recent observation as $\pi(a_t|o_t)$. The definition of the control policy is the most significant element that differentiates the IDM, Behavior Cloning, and AIDA. These differences are illustrated in the computation graphs in Fig. 5.1 and further described in the subsequent sections.



(a) IDM (b) AIDA (c) BC

Figure 5.1: Computation graphs for (a) IDM, (b) AIDA, and (c) neural network BC models. o = instantaneous observation, a = control action, h = complete interaction history, $\tilde{d}$ = desired distance headway, b = instantaneous belief, $\mathcal{G}$ = expected free energy, NN = neural network.

5.3.1 Intelligent Driver Model

The IDM [213] implements a control policy based on drivers' instantaneous observation of their own vehicle's speed v, relative speed to the lead vehicle Δv, and distance headway to the lead vehicle d, i.e., $\pi(a_t|o_t = \{v_t, \Delta v_t, d_t\})$. At each time step, the IDM computes a longitudinal acceleration to regulate the driver's vehicle towards a desired speed $\tilde{v}$ and desired distance headway $\tilde{d}$ using the following control rule:

$$a_t = a_{max}\left[1 - \left(\frac{v_t}{\tilde{v}}\right)^4 - \left(\frac{\tilde{d}}{d_t}\right)^2\right] \tag{5.1}$$

where the desired distance headway is defined as:

$$\tilde{d} = d_0 + v_t\tau - \frac{v_t\Delta v_t}{2\sqrt{a_{max}b}} \tag{5.2}$$

The IDM has the following parameters: a_{max} the maximum acceleration rate which can be implemented by the driver, d_0 the minimum allowable distance headway, τ the desired headway time, and b the maximum deceleration rate. While these parameters can be set manually by human designers, they usually depend on the road condition and vary with individual driver characteristics, e.g., the desired velocity and minimum distance headway. Thus, various methods have been proposed to calibrate model parameters from traffic data [214, 215].

A significant limitation of the IDM is that it cannot express certain types of behavior as a result of the control rule defined in (5.1) and (5.2). For example, it cannot express behavior due to uncertainty about the lead vehicle behavior and surrounding traffic and is limited to modeling behavior in non-conflict scenarios [199]. Incorporation of such behavior requires significant intervention from the model designers in adapting the control rule, e.g, modifying (5.1) and (5.2) to depend on additional inputs or "memory" mechanisms [198].

5.3.2 Behavior Cloning

BC refers to methods that train neural networks to learn policies from datasets of human car following behavior. The dataset, denoted with $\mathcal{D}$, is usually organized in the form of observation-action trajectories, i.e., $\mathcal{D} = \{o_{1:T}^{(i)}, a_{1:T}^{(i)}\}_{i=1}^{N}$, where N is the total number of trajectories and T is the length of each trajectory. The neural network parameterized policies depend on either the entire history h_t or the most recent observation o_t. Let us denote the policy parameters with θ, BC trains policies to maximize the expected log likelihood of the dataset trajectories:

$$\max_{\theta} \mathcal{L}(\theta) = \mathbb{E}_{o_{1:t}, a_{1:t} \sim \mathcal{D}} \left[\sum_{t=1}^{T} \log \pi_\theta(a_t | h_t) \right] \tag{5.3}$$

BC is simple to implement and more computationally efficient than comparative data-driven machine learning methods like reinforcement learning and online imitation learning. BC also does not require a high fidelity traffic simulation environment for training, which is necessary for reinforcement learning and online imitation learning. In contrast to rule-based policies, BC policies are more flexible and can express a much larger class of behaviors.

However, BC as a representative offline learning method has known disadvantages of being sensitive to the quantity and quality of training data and input features. The covariate-shift between the training dataset and the testing environment and neural network models' difficulty of extrapolating learned mechanisms to unseen inputs often cause BC models to overfit to the training dataset while producing poor control behavior during closed-loop testing (defined in section 5.3.8.2) [216, 205]. Furthermore, several studies have found that BC can be highly sensitive to input features [204, 217, 201]. Specifically, when a driver's previous control actions are used as input features to the trained policy, it is likely that the policy merely repeats those controls actions in closed-loop testing. This has been interpreted as a form of learning spurious correlations or causal confusion in machine learning, since driver controls at adjacent time steps are usually so similar that predicting previous controls can quickly minimize training error [217]. Because BC does not impose any structure on the policy, examining the failure modes of BC models is as challenging as

examining any other black-box neural network models.

Despite these shortcomings, BC, or variations of it, is a widely studied approach in developing automated vehicle algorithms and building simulated agents and environments for training them [210, 209]. It can produce high quality simulated behavior in practice when the training dataset is large and diverse, appropriate features are selected, and the neural network model is large and expressive enough [207, 201, 208] and thus it represents a valid data-driven modeling benchmark.

5.3.3 Active Inference Driving Agent

An active inference agent is defined by its internal generative model, which we implemented as a Partially Observable Markov Decision Process (POMDP). A POMDP describes a dynamic process in which the state of the environment $s_t \in \mathcal{S}$ evolves with driver actions, $a_t \in \mathcal{A}$, according to a probability distribution, $P(s_{t+1}|s_t, a_t)$, and generates observation signal, $o_{t+1} \in \mathcal{O}$, according to, $P(o_{t+1}|s_{t+1})$. In this work, we assume the observation signals are multivariate continuous variables and the states are discrete to represent probabilistic categorizations of the observation space (i.e., categorical perception [218]). At every time step, the active inference agent first makes inference about the hidden state of the environment upon receiving observations using Bayes' rule:

$$b_t(s_t) = \frac{P(o_t|s_t)P(s_t|b_{t-1}, a_{t-1})}{\sum_{s_t} P(o_t|s_t)P(s_t|b_{t-1}, a_{t-1})} \tag{5.4}$$

where $b_t(s_t) = P(s_t|h_t)$ denotes the agent's belief about the environment state given the observation-action history h_t and $P(s_t|b_{t-1}, a_{t-1}) = \sum_{s_{t-1}} P(s_t|s_{t-1}, a_{t-1})b(s_{t-1})$ is the prior predictive distribution based on the previous belief.

The active inference agent then selects control actions to minimize a criterion known as the (cumulative) expected free energy (EFE) [82]:

$$\mathcal{G}^*(b_t) = \min_{\pi} \mathbb{E}\left[\sum_{t}^{t+H} EFE(b_t, a_t) + \log \pi(a_t|b_t)\right] \tag{5.5}$$

where $H \leq \infty$ is a finite planning horizon. The EFE is defined as:

$$EFE(b_t, a_t) \triangleq \mathbb{E}[D_{KL}\left(b_{t+1}||\tilde{P}\right)] + \mathbb{E}[\mathcal{H}(o_{t+1})] \tag{5.6}$$

where $\tilde{P} := \tilde{P}(s_{t+1})$ defines the agent's preferred state distribution, $D_{KL}(\cdot||\cdot)$ denotes the Kullback-Leibler divergence — measuring the discrepancy between the current belief and the preferred state distribution — and $\mathcal{H}(\cdot)$ denotes Shannon entropy — measuring uncertainty about observations. These terms represent goal-seeking and information-seeking (epistemic) behavior respectively [32]. The first expectation in the EFE is taken with respect to:

$$P(o_{t+1}|b_t, a_t) = \sum_{s_{t+1}} P(o_{t+1}|s_{t+1})P(s_{t+1}|b_t, a_t) \tag{5.7}$$

and the second expectation in the EFE is taken with respect to $P(s_{t+1}|b_t, a_t)$.

Let $\mathcal{G}^*(b_t, a_t)$ be defined as:

$$\mathcal{G}^*(b_t, a_t) := EFE(b_t, a_t) + \log \pi(a_t|b_t) + \int P(o_{t+1}|b_t, a_t)\mathcal{G}^*(b_{t+1})d_{o_{t+1}} \tag{5.8}$$

Then the optimal policy has a closed-form expression [60]:

$$\pi(a_t|b_t) = \frac{e^{-\mathcal{G}^*(b_t,a_t)}}{\sum_{\tilde{a}\in\mathcal{A}} e^{-\mathcal{G}^*(b_t,\tilde{a})}} \tag{5.9}$$

Active inference has two important differences from the traditional notion of POMDP in operations research and reinforcement learning. First, both the generative model and the control objective are internal to the agent, meaning they can differ in substantial ways from the true environment generative model or a canonical notion of desired behavior, e.g., a "good driver" should always be centered in the lane. This has important implications as many human driving behaviors can be explained as inference in subjective or sub-optimal models [219]. Second, active inference makes an explicit distinction between pragmatic and epistemic behavior in its policy objective

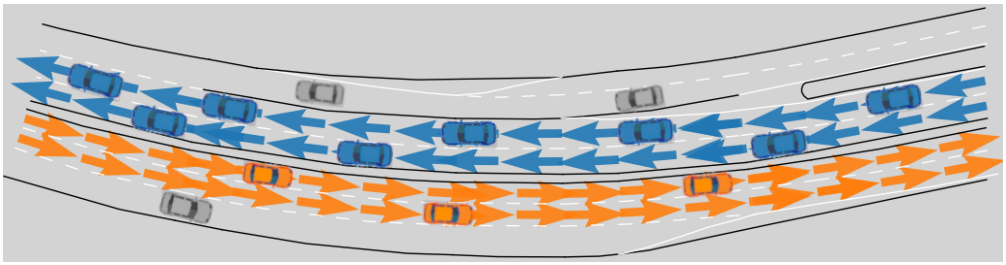


Figure 5.2: Top down view of the roadway explored in this analysis. We trained the models to emulate the behavior of the blue cars (traveling west) and evaluated the models' ability to predict behavior of the blue and orange cars (traveling east). Grey cars in the merging lanes were excluded.

according to the first and second terms in (5.6). This distinction supports adaptive behavior in unknown and uncertain environments [32, 175], e.g., traffic environments.

5.3.4 Dataset

We performed our analysis of the IDM, BC, and AIDA using the INTERACTION dataset [220], a publicly available driving dataset recorded using drones on fixed road segments in the USA, Germany, and China. The dataset provides a lanelet2 format map [221] and a set of time-indexed trajectories of the positions, velocities, and headings of each vehicle in the scene in the map's coordinate system at a sampling frequency of 10 Hz, and the vehicle's length and width for each road segment. The dataset contains a variety of traffic behaviors, including car following, free-flow traffic, and merges.

Due to our emphasis on car following behavior, we selected a subset of the data to include car following data from a two-way, seven-lane highway segment in China with a total distance of 175 m. We focused on vehicles in the middle two west-bound lanes shown in Fig. 5.2. We further filtered the remaining vehicles according to two criteria: 1) there was a lead vehicle with a maximum distance headway of 60 m, and 2) the ego vehicle was not performing a merge or lane change. We identified merging and lane change behavior using an automated logistic regression-based approach and validated the classifications with a manual review of a subset of trajectories. We also removed all trajectories with length shorter than 5 seconds, leaving a total of 1,254 trajectories with an average length of 14 seconds.

5.3.4.1 Feature Computation

The input features to the IDM are defined in (5.1) and (5.2). For BC and the AIDA, we used d and Δv but excluded v to prevent learning spurious correlations to ego speed or acceleration from past time steps reported in prior studies [217, 207, 201]. Furthermore, we included an additional feature τ^{-1} in BC and AIDA defined as the rate of change of the visual angle of the lead vehicle from the ego driver's seat position divided by the angle itself. τ^{-1} can be considered as a perceptual-control analog of inverse time-to-collision, a feature commonly used in driver modeling [204, 166, 164], with the difference of incorporating the width of the lead vehicle into feature computation and using quantities that can actually be observed by the driver. This is consistent with recent findings on the impact of optical expansion of the lead vehicle's image on driver longitudinal control behavior [162]. Furthermore, the inclusion of this feature puts the information contained in the inputs to BC and the AIDA on a similar level to the IDM, as the IDM implicitly accounts for time-to-collision in its desired distance headway computation in (5.2).

We computed all features in the Frenet frame (i.e., lane-centric coordinates [222]), by first transforming vehicle positions, velocities, and headings using the current lane center line as the reference path and then computing the features from the transformed positions and velocities. We obtained the drivers' instantaneous longitudinal control inputs (i.e., accelerations) from the dataset by differentiating the Frenet frame longitudinal velocities. For BC and the AIDA, we discretized the continuous control inputs into discrete actions using a Gaussian mixture model of 15 Gaussian components with mean and variance parameters chosen using the Bayesian Information Criteria [184].

5.3.5 Model Implementation

In this section, we describe our approach for parameterizing the IDM, BC, and the AIDA.

IDM. Following [223], we parameterized the IDM by treating the IDM policy as a conditional Gaussian distribution: $\pi(a_t|o_t = \{v_t, \Delta v_t, d_t\}) = \mathcal{N}(a_t|\mu_t, \sigma^2)$ with mean action μ_t and variance σ^2. The mean action μ_t is computed from the IDM rule defined in (5.1) and (5.2) by making the de-

sired speed $\tilde{v}$, minimum distance headway d_0, desired headway time τ, maximum acceleration rate a_{max} and maximum deceleration rate b adjustable parameters. The action variance σ^2 is assumed to be independent of the input features and also estimated from data.

BC. We implemented the BC model with two types of neural network policies: standard Multi-Layer Perceptron (MLP) networks and recurrent neural networks (RNN) following [204]. The MLP network takes as input the observation vector (normalized by training set statistics) and outputs a probability distribution over the discrete control actions. The RNN addresses the possibility that driver behavior may be influenced by the full observation history rather than just the most recent observation. For the RNN, we combined a Gated Recurrent Unit (GRU) network [224] with a MLP network, where the GRU network compresses the observation history into a fixed length vector, which is then transformed into the action distribution by the MLP network.

AIDA. We modeled the discrete state transition probabilities $P(s_{t+1}|s_t, a_t)$ and the desired state distribution $\tilde{P}(s_t)$ of the AIDA using categorical distributions parameterized by their logits. We parameterized the observation distributions $P(o_t|s_t)$ using Normalizing Flow, a flexible class of neural network-based density estimator [225, 226]. This provides the AIDA with adequate flexibility in modeling complex and nonlinear observation sequences and associating observed actions with agent beliefs. Normalizing Flow uses invertible neural networks to transform simple distributions, e.g., Gaussian distributions, into complex and correlated distributions while maintaining the tractability of likelihood evaluation and sampling. In this work, we used a single, shared Inverse Autoregressive Flow [227] to transform a set of conditional Gaussians with mean vector $\mu(s_t)$ and covariance matrix $\Sigma(s_t)$. We modeled a distribution over the agent's planning horizon using a Poisson rate parameter and used the QMDP method [42, 228] as a closed-form approximation of the cumulative expected free energy in (5.8). We approximately computed the entropy of the state-conditioned observation distributions required in the EFE calculation using the entropy of the Normalizing Flow base distributions. In subsequent sections, we refer to the transition and observation parameters with θ_1, the desired state distribution and planning horizon parameters with θ_2, and the combined parameters with $\theta = \{\theta_1, \theta_2\}$.

We provide additional implementation details in Appendix C. Our software implementation is publicly available at [229].

5.3.6 Parameter Estimation

We estimated the parameters of the IDM, BC, and AIDA by maximizing the expected log likelihood of driver control inputs from the dataset under the corresponding control policy, i.e., (5.3). This procedure for the AIDA differs slightly from the IDM and BC by requiring a nested step. Between each parameter update in the nested procedure, we first computed the sequence of beliefs given the observation-action history using (5.4) and the optimal belief-action policy using (5.9). We then evaluated the log likelihood as a function of the computed beliefs:

$$\max_{\theta} \mathbb{E}_{o_{1:T},a_{1:T}\sim\mathcal{D}}\left[\sum_{t=1}^{T-1}\log \pi_\theta(a_t|b_{t,\theta_1})\right] \quad \text{s.t.} \quad \pi_\theta(a_t|b_t) = \frac{e^{-\mathcal{G}^*_{t,\theta}(b_t,a_t)}}{\sum_{\tilde{a}_t\in\mathcal{A}} e^{-\mathcal{G}^*_{t,\theta}(b_t,\tilde{a}_t)}} \tag{5.10}$$

While (5.10) allows us to learn task-relevant beliefs in active inference agents as it depends on both θ_1 and θ_2, the parameters are fundamentally unidentifiable since there are potentially infinite sets of θ with the same likelihood [109, 230, 141]. This is because, for example, the estimation algorithm cannot differentiate between drivers who desire a small distance headway and drivers who believe the distance headway will increase to desired levels in subsequent time steps. A possible consequence of this is learning an environment model that deviates significantly from the reality, which leads to a large number of crashes or inactions as a result of the agent not being able to recognize the actual environment state [231].

In order to constrain the hypothesis space and avoid configurations of θ that are incompatible with real-world constraints, we designed a data-driven prior distribution $P(\theta)$ encoding likely configurations of θ. Specifically, the prior is defined as $P(\theta) = P(\theta_1)P(\theta_2|\theta_1)$, where:

$$P(\theta_1) \propto \exp\left(\lambda \mathbb{E}_{o_{1:T},a_{1:T}\sim\mathcal{D}}\left[\sum_{t=1}^{T}\log P_{\theta_1}(o_t|h_{t-1},a_{t-1})\right]\right) \tag{5.11}$$

with hyperparameter λ controlling how much the prior distribution prefers model accuracy, mea-

sured by expected log likelihood of observations. We let $P(\theta_2|\theta_1)$ be a uniform distribution. In our experiments, we only compute the Maximum A posteriori (MAP) estimate of the Bayesian model by converting the prior into the following loss function added to the objective in (5.10):

$$\mathcal{L}(\theta_1) = \lambda \mathbb{E}_{o_{1:T}, a_{1:T} \sim \mathcal{D}} \left[\sum_{t=1}^{T} \log P_{\theta_1}(o_t | h_{t-1}, a_{t-1}) \right] \tag{5.12}$$

To prevent learning unreasonably large observation variance as a result of the observation entropy term in (5.6), another symptom previously reported in [231], we applied a penalty on the l^2 norm of the observation covariance parameters.

Using these prior loss functions, the AIDA MAP estimate can be written as:

$$\theta^{\text{MAP}} = \arg\max_{\theta} \sum_{t=1}^{T} \mathbb{E}_{o_{1:T}, a_{1:T} \sim \mathcal{D}} \left[\log \pi_\theta(a_t | b_{t,\theta_1}) + \lambda_1 \log P_{\theta_1}(o_t | h_{t-1}, a_{t-1}) \right] + \lambda_2 \sum_{s} ||\Sigma_{\theta_1}(s)||^2 \tag{5.13}$$

5.3.7 Model Selection

We trained each model with 15 random seeds controlling model parameter initialization and dataset mini-batch iteration orders. To select the hyperparameters for the AIDA, we first set $\lambda_2 = 0.1$ since it's sufficient to mitigate overly large covariances. We then trained the model for $\lambda_1 = [0.2, 1, 4]$ and selected $\lambda_1 = 1$ as it best trades off environment model accuracy and agent behavior predictive performance (with criteria described in the next section).

5.3.8 Model Evaluation and Comparison

We evaluated and compared our models' ability to generate behavior similar to the human drivers in the dataset using both open-loop offline predictions and closed-loop online simulations. In both cases, we evaluated the models on two different held-out testing datasets. The first dataset includes vehicles from the same lanes as the training dataset. This dataset tests whether the models can generalize to unseen vehicles in the same traffic condition. We obtained this dataset by dividing all selected trajectories in the westbound lanes using a 7-3 train-test ratio. The second dataset

includes vehicles from the top two eastbound lanes in Fig. 5.2. This dataset tests whether the models can generalize to unseen vehicles in novel traffic conditions, since the traffic in the eastbound lanes have on average higher speed and less density. We refer to these two datasets as *same-lane* and *new-lane*, respectively. We randomly selected 100 trajectories with a minimum length of 10 seconds from the same-lane dataset and 75 trajectories with a minimum length of 5 seconds from the new-lane dataset for testing.

5.3.8.1 Offline Evaluation

The goal of the offline evaluation was to assess each model's ability to predict a driver's next action based on the observation-action history recorded in the held-out testing dataset. This task evaluates the models' ability to be used as a short-horizon predictor of other vehicles' behavior in an on-board trajectory planner [232]. We measured a model's predictive accuracy using Mean Absolute Error (MAE) of the predicted control inputs (unit=m/s^2) on the held-out testing datasets. For the IDM, we calculated the predicted control inputs by sampling from the Gaussian policy. For BC and the AIDA, we first sampled a discrete action from the action distribution predicted by the models and then sampled from the corresponding Gaussian component in the Gaussian mixture model used to perform action discretization.

5.3.8.2 Online Evaluation

Rather than predicting instantaneous actions, the goal of the online evaluation was to assess the models' ability to generate trajectories similar to human drivers such that they can be used as simulated agents in automated vehicle training and testing environments [209]. This is fundamentally different from offline predictions because the models need to choose actions based on observation-action history generated by its own actions rather than those stored in the fixed, offline dataset. This can introduce significant covariate shift [205] sometimes resulting in situations outside of the model's training data, which can lead to poor action selection.

We built a single-agent simulator where the ego vehicle's longitudinal acceleration is controlled by the trained models and its lateral acceleration is controlled by a feedback controller for lane-

centering. The lead vehicle simply plays back the trajectory recorded in the dataset. Other vehicles do not have any effect on the ego vehicle, given our observation space does not contain other vehicle related features.

Following [210], We measured the similarity between the generated trajectories and the true trajectories using the following metrics:

1. Average deviation error (ADE; unit=m): deviation of the Frenet Frame longitudinal position from the dataset averaged over all time steps in the trajectory.

2. Lead vehicle collision rate (LVCR; unit=$\%$): percentage of testing trajectories containing collision events with the lead vehicle. A collision is defined as an overlap between the ego and lead vehicles' bounding boxes.

5.3.8.3 Statistical Evaluation

Following the recommendations in [233, 234] for evaluating learned control policies, we represented the central tendency of a model's offline prediction and online control performance using the interquartile mean (IQM) of the offline MAEs and online ADEs. Note however for collision rate, we compute the regular mean instead of IQM to account for the collision rate lower bound of 0. The IQMs are computed by 1) ranking all tested trajectories by their respective performance metrics and 2) computing the mean of the performance metrics ranked in the middle 50%. To compare the central performance difference between the AIDA and baseline models, we performed two-sided Welch's t-tests with 5 percent rejection level on the MAE-IQM and ADE-IQM values computed from different random seeds with the assumption that the performance distributions between two models may have different variances [233, 234].

5.4 Results and Discussion

5.4.1 Offline Performance Comparison

Fig. 5.3 shows the offline evaluation results for each model with the model type on the x-axis and the IQMs of acceleration prediction MAEs averaged across the testing dataset on the y-axis.

The color of the points in the figure represents the testing condition and each point corresponds to a random seed's result. The points are randomly distributed around each x-axis label for clarity. Dispersion on the y-axis indicates sensitivity in the model to initial training conditions. The plot illustrates that the AIDA had the lowest MAE-IQM in the same-lane tests, followed by BC-RNN, BC-MLP, and IDM. The corresponding pairwise Welch's t-test results in Table 5.1 show that these differences are significant. In the new-lane tests, both the AIDA and neural network BC models significantly outperformed IDM. The AIDA performance has higher variance than BC models, however the difference in the central tendency was not significant. These results show that in the current car following setting, the AIDA and BC generalized better to the new-lane scenario than the IDM, mostly likely due to the IDM rules being unable to adapt to different traffic speed and density than the training dataset. The stronger predictive performance in the AIDA and BC-RNN in the same-lane data can be attributed to the fact that driver acceleration actions depend on the full history of past observations rather than just the most recent observation, which can be modeled by the recurrent structure of the AIDA and BC-RNN. The figure also shows that for the same-lane tests, the AIDA had more variance across the random seeds compared to other models, suggesting that it is the most sensitive to local optima in the training process.

Table 5.1: Two-sided Welch's t-test results of offline MAE-IQM against baseline models. Asterisks indicate statistical significance with $\alpha = 0.05$.

Baseline	Comparison	t(df=14)	p-value
IDM	same-lane	t=37.58	p<0.001*
BC-MLP	same-lane	t=32.38	p<0.001*
BC-RNN	same-lane	t=17.31	p<0.001*
IDM	new-lane	t=33.21	p<0.001*
BC-MLP	new-lane	t=0.35	p=0.73
BC-RNN	new-lane	t=-0.12	p=0.90

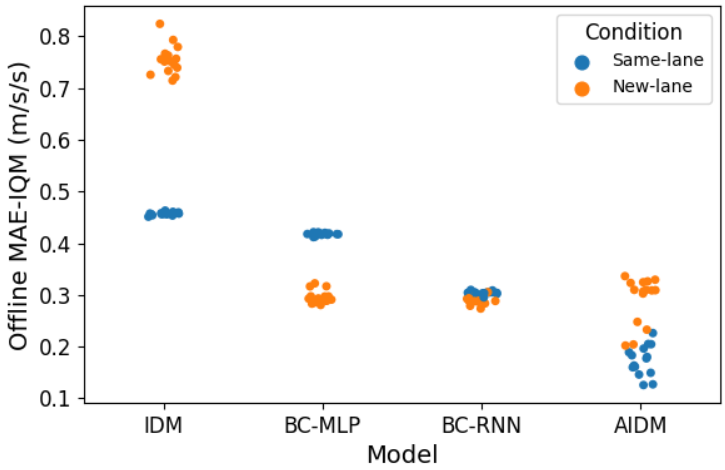


Figure 5.3: Offline evaluation MAE-IQM. Each point corresponds to a random seed used to initialize model training and its color corresponds to the testing condition of either same-lane or new-lane.

5.4.2 Online Performance Comparison

Fig. 5.4 shows the IQM of each model's ADEs from data set trajectories in the online evaluations using the same format as the offline evaluation results. In the same-lane testing condition, all models had ADE-IQM values between 1.8 m and 2.8 m, which is less than the length of a standard sedan ($\approx$ 4.8 m; [235]). Among all models, BC-MLP achieved the lowest ADE values for both the same-lane and new-lane conditions, followed by the AIDA, IDM, and BC-RNN. Furthermore, both the AIDA and BC models achieved lower ADE-IQM in the new lane settings compared to the same-lane setting, however the IDM achieved higher ADE-IQM in the new-lane setting. The Welch's t-test results in Table 5.2 show that AIDA's online test performances are significantly different from all baseline models in both the same-lane and new-lane settings ($P \leq 0.01$). These findings confirm that the AIDA and BC models generalized better to the new-lane setting than the IDM and suggest that the AIDA's average online trajectory-matching ability is significantly better than IDM and BC-RNN, although BC-MLP is significantly better than the AIDA.

Fig. 5.5 shows the lead vehicle collision rates for each random seed and model using the same format as Fig. 5.4. The figure illustrates that in the same-lane condition, the random seeds for

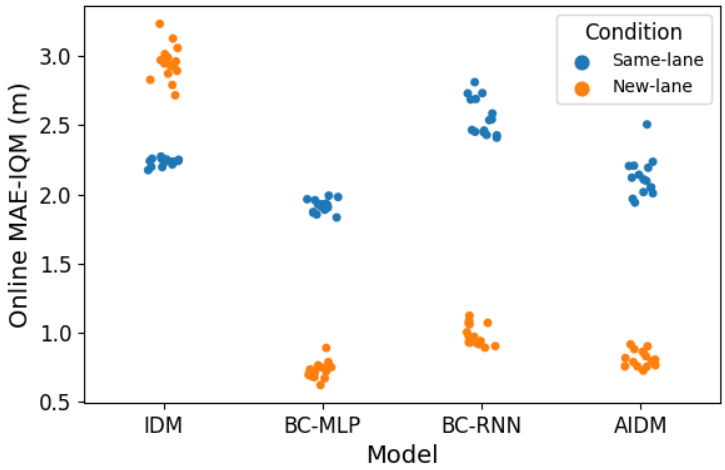


Figure 5.4: Online evaluation ADE-IQM. Each point corresponds to a random seed used to initialize model training and its color corresponds to the testing condition of either same-lane or new-lane.

Table 5.2: Two-sided Welch's t-test results of online ADE-IQM against baseline models. Asterisks indicate statistical significance with $\alpha = 0.05$.

Baseline	Comparison	t(df=14)	p-value
IDM	same-lane	t=3.05	p<0.01*
BC-MLP	same-lane	t=-5.46	p<0.001*
BC-RNN	same-lane	t=8.73	p<0.001*
IDM	new-lane	t=58.18	p<0.001*
BC-MLP	new-lane	t=-3.77	p<0.001*
BC-RNN	new-lane	t = 6.87	p<0.001*

BC-MLP, BC-RNN, and the AIDA had more collisions than the IDM (0% collision rate across all seeds). In particular, BC-RNN and the AIDA had substantial differences across random seeds compared to the other models. However, the minimum collision rates for BC-MLP, BC-RNN, and the AIDA were consistent (less than or equal to 1%). In the new-lane condition, the collision rate was 0% for all four models. The higher collision rates in the same-lane data are likely due to the traffic density and complexity, which were higher in the same-lane condition compared to the new-lane condition.

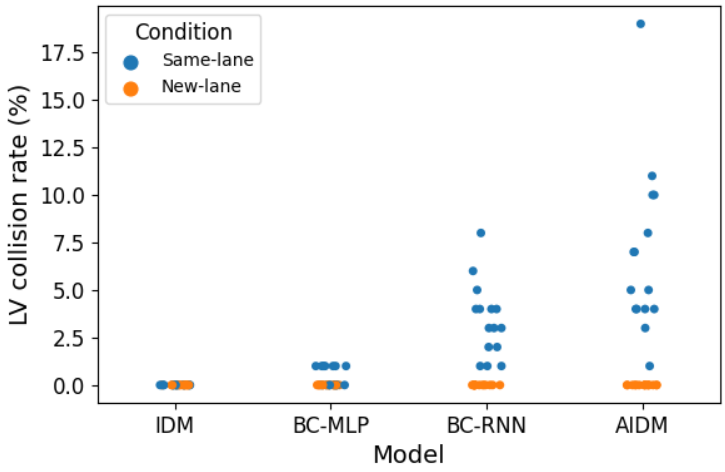


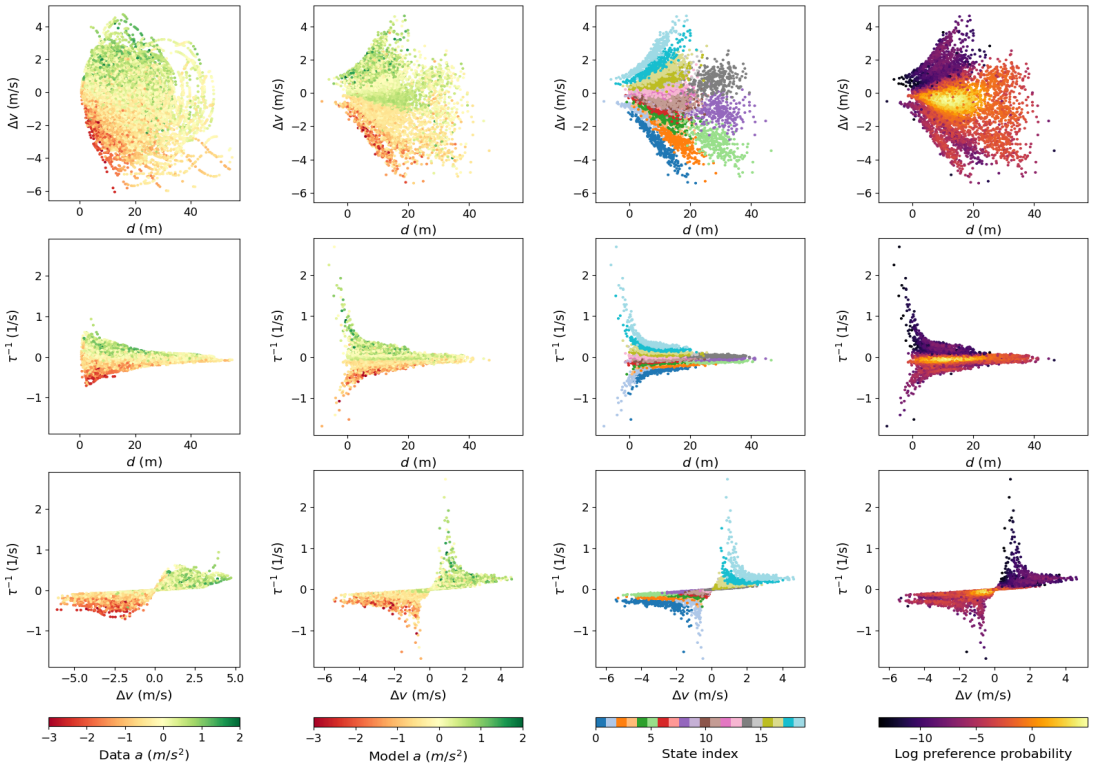
Figure 5.5: Lead vehicle collision rate in online evaluation. Each point corresponds to a random seed used to initialize model training and its color corresponds to the testing condition of either same-lane or new-lane.

5.4.3 AIDA Interpretability Analysis

The previous sections suggest that the AIDA can capture driver car following behavior significantly better than the IDM and comparably to data-driven BC models. However, the findings have yet addressed the interpretability of the AIDA. While there is no established metric for model interpretability, Räukur et. al. [195] recommend assessments based on the easiness of comprehending the connection between model input and output and tracing model predictive errors to internal model dynamics. Given that the AIDA's decisions are emitted from a two-step process, i.e., (1) forming beliefs about the environment and (2) selecting control actions that minimize free energy, the model's success at the car following task depends on the two sub-processes both independently and jointly. We examined the AIDA's learned input-output mechanism by visualizing its independent components (i.e., the observation, transition, and preference distributions) and verified them against expectations guided by driving theory [178, 175, 236]. We then examined the joint belief-action process by replaying the AIDA beliefs and diagnosing its predictions of recorded human drivers in the offline setting and its own decisions in the online setting.

5.4.3.1 *Independent Component Interpretability*

Initial insights into the model input and output connections can be gained by visualizing the AIDA components, specifically its policy (Fig. 5.6b), observation distribution (shown in Fig. 5.6c), and preference distribution (Fig. 5.6d). These figures show 200 random samples from each state of the AIDA's state-conditioned observation distribution, $P(o|s)$, plotted on each pair of observation modalities. Color is used to highlight relevant quantities of interest. We further used samples drawn from the INTERACTION dataset, plotted in Fig. 5.6a and colored by the recorded accelerations, to facilitate interpreting the AIDA samples.



(a) (b) (c) (d)

Figure 5.6: Visualizations of the dataset and AIDA model components. In panel (a), we plotted observations sampled from the dataset. In panels (b), (c), and (d) we sampled 200 points from the AIDA's state conditioned observation distributions and plotted the sampled points for each pair of observation feature combinations. The points in each panel are colored by: (a) accelerations from the dataset, (b) the AIDA's predicted accelerations upon observing the sampled signals from a uniform prior belief, (c) state assignments (d) log probabilities of the preference distribution.

Fig. 5.6b illustrates the observation samples by the model's chosen control actions. The top chart shows the samples using distance headway (d; x-axis) by relative velocity to the lead vehicle (Δv; y-axis), the middle chart shows distance headway by τ^{-1}, and the bottom chart shows relative velocity by τ^{-1}. The shape of the sampled points matches the contour of the empirical dataset (Fig. 5.6a), particularly in the middle and bottom visualizations, which suggests that the model's learned observation model aligns with the recorded observations in the dataset. Darker green and red colors correspond to larger acceleration and deceleration magnitudes, respectively, and light yellow color corresponds to near zero control inputs. The color gradient at different regions in Fig. 5.6b is consistent with that of the empirical dataset shown in Fig. 5.6a. This shows that the model learned a similar observation to action mapping as the empirical dataset. The mapping can be interpreted as the tendency to choose negative accelerations when the relative speed and τ^{-1} are negative and the distance headway is small, and positive accelerations in the opposite case. Furthermore, the sensitivity of the red and green color gradients with respect to distance headway shows that the model tends to accelerate whenever there is positive relative velocity, regardless of the distance headway. However, it tends to input smaller deceleration at large distance headway for the same level of relative speed.

Fig. 5.6c shows the observation samples colored by their associated discrete states. The juxtaposition of color clusters in the top panel shows that the AIDA learned to categorize observations by relative speed and distance headway and its categorization for relative speed is more fine-grained at small distance headways and spans a larger range of values. The middle and bottom panels show that its categorization of relative speed is highly correlated with τ^{-1} as the ordering of colors along the y-axis is approximately the same as in the top panel. The middle and bottom panels show that the AIDA's categorization of high τ_1 magnitude states (blue and cyan clusters) have a larger span than that of low τ^{-1} magnitude states. These patterns further establish that the AIDA has learned a representation of the environment consistent with the dataset. At the same time, it can be interpreted as a form of satisficing in that the model represents low urgency large distance headway states with less granularity [237].

Fig. 5.6d shows the observation samples by the log of its preference probability, $\tilde{P}(o) = \sum_s \tilde{P}(s)P(o|s)$, where higher preference probability (i.e., desirability) corresponds to brighter colors (e.g., yellow) and lower desirability corresponds to darker colors (e.g., purple). The figure shows that the highest preference probability corresponds to observations of zero τ^{-1}, zero relative velocity, and a distance headway of 18 m (see the center region of the middle chart, and the yellow circle at the left-center of the top chart). This aligns with the task-difficulty homeostasis hypothesis that drivers prefer states in which the crash risk is manageable [178] and not increasing. It is also consistent with the observed driver behavior in Fig. 5.6a where drivers tend to maintain low accelerations (light yellow points) within the same regions.

Overall, these results show a clear mapping between the AIDA's perceptual (Fig. 5.6c) and control (Fig. 5.6d and 5.6b) behavior that is both consistent with the observed data and straightforwardly illustrated using samples from the fitted model distributions. This mapping facilitates predictions of the AIDA's reaction to observations without querying the model, which is an important dimension of interpretability in real world model verification [195].

5.4.3.2 Joint Model Interpretability

While the previous analysis illustrates the interpretability of individual model components, the interaction between components introduces additional challenges for overall model interpretability. To address this, we analyzed two same-lane scenarios where the AIDA made sub-optimal decisions in the model testing phase — one from the offline evaluations where the AIDA's predictions had the largest MAE and one from the online evaluations where the AIDA generated a rear-end collision with the lead vehicle. We first visualized the AIDA's beliefs (computed by (5.4)) and policies (computed by (5.9)) as the model generated actions and then used those visualizations to demonstrate how the transparent input-output mechanism in the AIDA can be used to mitigate the sub-optimal decisions.

The chosen offline evaluation trajectory is visualized in Fig. 5.7. The left column charts show the data of the three observation features over time. The right column charts show the time-varying ground truth action probabilities over time (top), action probabilities predicted by the AIDA over

time (middle), and environment state probabilities $P(s|h)$ inferred by the AIDA over time (bottom). In the right-middle and right-bottom charts, the action and belief state indices are sorted by the mean acceleration and τ^{-1} value of each state to facilitate alignment with the left and top-right charts. We labeled the actions by the corresponding means but not the belief states because they represent multi-dimensional observation categorizations (see Fig. 5.6c). The bottom-right chart shows that the inferred belief patterns closely followed the observed relative speed and τ^{-1} in the left-middle and left-bottom charts with high precision, i.e., close to probability of 1. The predicted action probabilities in the right-middle chart followed the trend of the ground truth actions, however, they exhibited substantially higher uncertainty at most time steps and multi-modality at $t = 1$ s and $t = 12$ s, where one of the predicted modes coincided with the true actions. Given the inferred beliefs were precise, uncertain and multi-model actions were likely caused by inter-driver variability in the dataset, where drivers experienced similar belief states but selected different actions. Alternatively, this uncertainty may be caused by drivers having highly different beliefs after experiencing similar observations, where a simple policy would be sufficient to predict their actions. In either case, the error in AIDA predictions can be attributed to inconsistency between the belief trajectories and action predictions.

The chosen online evaluation trajectory which resulted in a rear-end collision with the lead vehicle is shown in Fig. 5.8 plotted using the same format as Fig. 5.7. The duration of the crash event is highlighted by the red square in the bottom-left chart, where the sign of τ^{-1} values instantly inverted when overlapping bounding boxes between the ego and lead vehicle first occurred and eventually ended. The AIDA initially made the correct and precise decision of braking, however, its predictions for high magnitude actions became substantially less precise prior to the collision ($t > 1$ s; see right middle chart). This led to the model failing to stop fully before colliding with the lead vehicle. The belief pattern shows that the AIDA tracked the initial decreasing values of relative speed and τ^{-1} but did not further respond to increasing magnitude of τ^{-1} 3 seconds prior to the crash (starting at $t = 1.6$ s). These findings show that the model exhibited the correct behavior of being "shocked" by out-of-sample near-crash observations, however, the learned categorical belief

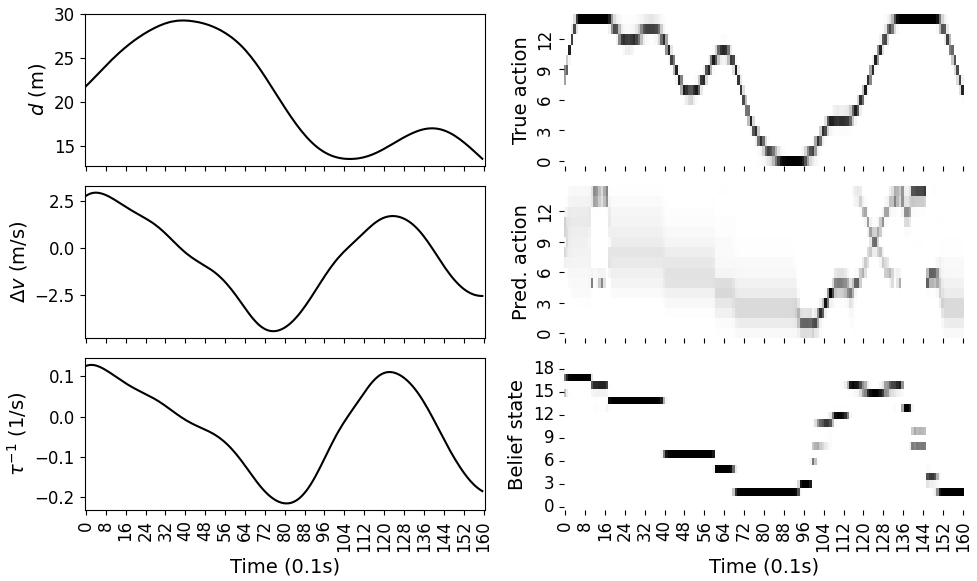


Figure 5.7: Visualizations of a same-lane offline evaluation trajectory where the AIDA had the highest prediction MAE. The charts in the left column show distance headway, relative speed, and τ^{-1} signals observed by the model over time. The binary heat maps in the right column show the ground truth action probabilities (top), action probabilities predicted by the AIDA (middle), and the corresponding belief states (bottom) over time (x-axis), where darker colors correspond to higher probabilities. The belief state and action indices are sorted by the mean τ^{-1} and acceleration value of each state, respectively.

representation was not able to extrapolate beyond the data from the crash-free INTERACTION dataset.

The analysis of the near-crash AIDA beliefs suggests that editing the AIDA's learned environment dynamics model (i.e., the transition and observation distributions) to properly recognize near-crash observation signals can likely avoid the current crash. To demonstrate the utility of being able to make precise model-editing decisions based on the interpretability analysis, we tested a modification of the AIDA by replacing its learned dynamics model with a physics-based dynamics model assuming constant lead vehicle velocity in the model predictions. Although the physics-based dynamics model does not capture the stochasticity in the lead vehicle behavior, it is sufficient for mitigating the current crash given its ability to accurately predict near-crash observations. We evaluated this new model in the same online testing scenarios as the AIDA, where the

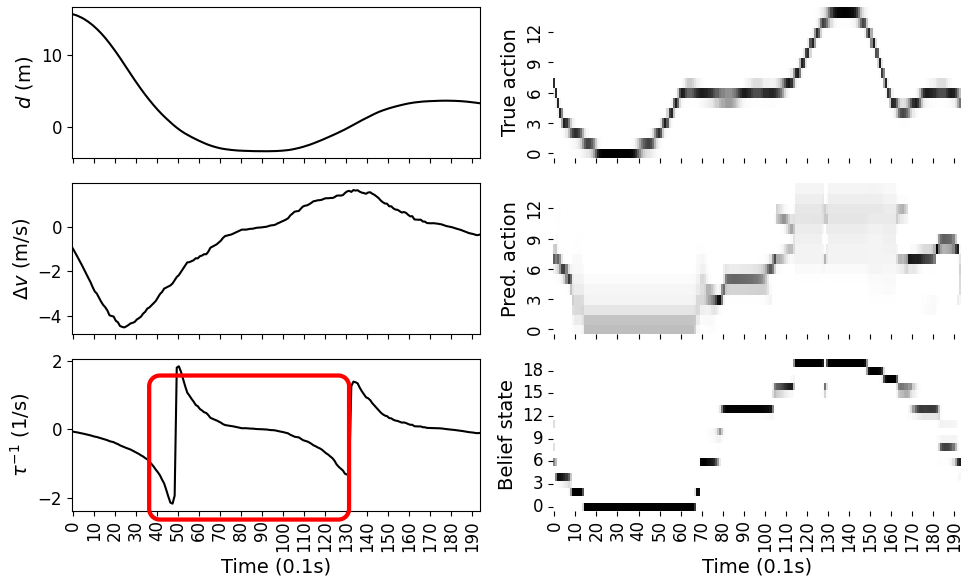


Figure 5.8: Visualizations of a same-lane online evaluation trajectory where the AIDA generated a rear-end collision with the lead vehicle. This figure shares the same format as Fig. 5.7. The red square in the bottom-left chart represents the duration of the rear-end crash event where the vehicle controlled by the AIDA had an overlapping bounding box with the lead vehicle.

control actions were generated from a model-predictive controller (MPC [238]) using the AIDA's preference distribution as the reward function (for detailed implementation seed Appendix C.3). The AIDA-MPC mitigated all crashes when deployed in the same scenarios as the AIDA as our analysis predicted. However, it generated substantially more high-ADE trajectories than the AIDA, most likely due to the lack of representation of lead vehicle stochasticity.

The analyses in this section show that the decision making structure in the AIDA enables modelers to reason about the training dataset's effect on the learned model behavior. To the best of our knowledge, this analysis is not possible with neural network BC models using existing interpretability tools. We also showed how this understanding can be used to edit parts of the model to achieve desired safety criteria.

5.5 General Discussion

In this article, we introduced and evaluated a novel active inference model of driver car following behavior (AIDA). The proposed AIDA significantly outperformed the IDM and neural network

BC models in offline predictions in the same-lane condition and outperformed the IDM while performing similarly to BC models in the new-lane condition. Additionally, the AIDA achieved significantly lower average deviation error than the IDM and BC-RNN in the online control settings. However, the results showed that the AIDA was sensitive to initial training conditions, which resulted in higher rates of lead vehicle collisions in the same-lane condition compared to the IDM and BC-MLP. While BC had comparable or better performance than the AIDA in action prediction and control, the AIDA is substantially more interpretable than BC models. In contrast to approximate explanatory methods for BC neural networks, we showed that the AIDA's decision making process can be directly accessed by sampling and visualizing the AIDA distributions. Further, we illustrated how the AIDA's joint belief and action trajectories could be used to understand model errors and correct them. This level of understanding and diagnostic analysis is central to real world model inspection and verification which are essential components of interpretability [239, 195].

These results partially confirm our hypothesis that balancing the relative strengths of rule-based and data-driven models, specifically using the active inference framework, results in better predictions of driver behavior and more nuanced understanding of driver cognitive dynamics during car following. In contrast to fixed rule-based models like the IDM, the AIDA can incorporate additional "rules" in its state and policy priors while maintaining the flexibility provided by its probabilistic representation. In contrast to purely data-driven models, learning in the AIDA is constrained by its probability distributions and structure. This balance preserves interpretability but still allows the model to be flexible to new data. Our findings here suggest that this flexibilty comes at a cost of sensitivity to local optima in the training process as evidenced by the collision rates across random seeds in online evaluations. Further, our findings suggest that the AIDA, like other data-driven approaches, may be limited by the scope of the data used in training (e.g., the crash limitation illustrated in Fig. 5.8).

Our findings here extend prior applications of active inference theory in driving and driver models and illustrate the value of rule-based modeling. Engström and colleagues [175] presented active inference as a general theory of driving behavior with qualitative illustrations, highlighting

the need to separate pragmatic (risk) and epistemic (uncertainty) behavior and relaxing the requirement of a strictly accurate environment model among human drivers. Portions of this theory have been enacted in other driver models including [168, 15, 202]. The model in [202] includes the concept of balancing rule-based and data-driven models, but the focus is primarily on physical concepts rather than psychological concepts in the AIDA. The model presented by Pekkanen et al. [168] includes an attention mechanism driven by the uncertainty of desired actions. The desired actions were computed using the IDM and action uncertainty was obtained by propagating state uncertainty computed from a Bayesian filter. The most notable difference between Pekkanen et al.'s model and the AIDA is that their model assumes an accurate environment model and uses the IDM to generate behavior. Our results show that an integrated perception-action system is important to the AIDA's trajectory-matching performance. However, we did not investigate epistemic behavior in the model due to the simplicity of the car following task. The AIDA posed here also extends our prior work [15] to model fine-grained longitudinal control, validate that model against established benchmarks, and provide a more detailed interpretability analysis.

In addition to the contributions to driver modeling, this work extends research on human perception and control modeling. Our simultaneous estimation of human preference, understanding of environment dynamics (i.e., transition probabilities), and perceptual uncertainties (i.e., observation probabilities), and use of data from a complex driving environment differentiate this work from [124, 109, 105]. Our findings here suggest that the AIDA can be extended to complex environments successfully, although it is sensitive to training data and model parameterization. Our use of a data-driven prior distribution, i.e.,(5.11), to prevent estimating transition and observation parameters that are highly inconsistent with actual traffic dynamics and reduce unidentifiability is also novel and differentiates this work from [228] and [240]. Our visualizations of model preference and beliefs in Fig. 5.6d and Fig. 5.7-5.8 show that the proposed data-driven prior leads to preference and dynamics estimation consistent with the observed data and driver behavior theories.

Our work is limited by the following aspects. First, we have assumed three driver observation modalities: distance headway, relative speed, and τ^{-1} with respect to the lead vehicle. However,

human drivers are known to monitor other surrounding vehicles while driving [8] and to have broader visual sampling [241]. Second, our parameterization of discrete states has limited the expressivity of the model and prevented inductive biases such as the smoothness of physical dynamics from being encoded. The limited dataset coverage, e.g., the lack of crashes, prevented the learned dynamics from generalizing to some out-of-distribution scenarios. The combination of model and data insufficiency led to the difficulty of recognizing near-crash states and resulted in substantially more lead vehicle crashes than BC-MLP and the IDM. Third, since the INTERACTION dataset was collected on highways, there likely exists considerable heterogeneous driving behavior. This is shown in the uncertain and multi-modal predictions in Fig. 5.7 as the model had to explain drivers who took different actions upon observing similar signals. While we anticipate incorporating additional observations and higher state space dimension and application to alternative driving scenarios to be easy under the current model formulation, doing so would impose additional requirements on dataset quality and diversity. We thus recommend future work to consider general methods for incorporating domain knowledge in more expressive generative models to combat dataset limitations and modeling heterogeneity in naturalistic driver behavior. The results here suggest that these extensions may alleviate many of the current model limitations.

5.6 Conclusions

We proposed a novel active inference model of driver behavior (AIDA). Using car following data, we showed that the AIDA significantly outperformed the rule-based IDM on all metrics and performed comparably with the data-driven neural network benchmarks. Using an interpretability analysis, we showed that the structure of the AIDA provides superior transparency of its input-output mechanics than the neural network models. Future work should focus on training with data from more diverse driving environments and examining model extensions that can capture heterogeneity across drivers.

6. UNDERSTANDING THE ROBUSTNESS OF BAYESIAN THEORY OF MIND*

6.1 Summary

The goal of this chapter is to develop a theoretical understanding of the observation from the previous chapter that the active inference model estimated using theory of mind inference substantially outperformed an RNN-based behavior cloning model, although both models belong to the recurrent model class and have similar representation capacity. Specifically, we aim to understand the performance advantage of control policies obtained from theory of mind inference. As previously discussed, the main difference between TOM and regular learning from demonstration techniques is that TOM performs belief inference, i.e., it tries to infer the demonstrator's *internal* dynamics model of the environment, simultaneously with reward inference. We show in this chapter that if we believe the agent has an accurate model of the environment, encoded using a family of priors parameterized by the log likelihood of dataset transitions (in fact the same prior introduced in Chapter 3), then TOM is transformed into a class of robust inference problem where it tries to find the worst-case dynamics outside the training data and the learner is be encouraged to stay close to the data distribution. Unlike existing offline RL and IRL methods which keep the learner policy close to the data distribution using ad hoc uncertainty-based penalties, this is naturally achieved under the TOM framework. We propose a set of algorithms following the TOM principle and show that they outperform state-of-the-art offline IRL methods on high-dimensional continuous control tasks.

6.2 Introduction

Inverse reinforcement learning (IRL) is the problem of extracting the reward function and policy of a value-maximizing agent from its behavior [102, 110]. IRL is an important tool in domains where manually specifying reward functions or policies is difficult, such as in autonomous driv-

*Reprinted, with permission, from Wei, R., Zeng, S., Li, C., Garcia, A., McDonald, A., & Hong, M. (2023, June). Robust Inverse Reinforcement Learning Through Bayesian Theory of Mind. In First Workshop on Theory of Mind in Communicating Agents. Copyright 2023 by the author(s).

ing [119], or when the extracted reward function can reveal novel insight about a target population, such as in biology and economics [242, 104]. Furthermore, IRL has been argued as a central mechanism of human theory of mind [14] and one of the main approaches for building value-aligned artificial intelligence [243]. However, wider application of IRL faces two interrelated algorithmic challenges: 1) having access to the target deployment environment or an accurate simulator thereof and 2) robustness of the learned policy and reward function due to the covariate shift between the training and deployment environments [216, 205, 244].

To tackle the first challenge, recent IRL research has focused on the *offline* setting, where only a fixed dataset is provided as opposed to the target environment or an accurate simulator [245, 246, 247, 121, 122]. Model-free approaches to offline IRL attempt to directly estimate expert reward and policy without building an explicit model of the environment dynamics [245, 246, 247]. In contrast, model-based offline IRL approaches estimate a dynamics model from the offline dataset [122, 121, 248, 249]. Both model-free and model-based offline IRL suffer from covariate shift due to error in either the policy or the dynamics model. However, model-based approaches, which will be our focus, hold more promise due to the ability to generate synthetic data and leverage model generalization.

A notable class of these model-based offline IRL methods estimate the dynamics and reward in a two-stage, *decoupled* fashion [121, 122, 248, 249]. In the first stage, a dynamics model is estimated from the fixed dataset. Then, parameters of the dynamics model are fixed while training the reward and policy in the second stage. To overcome covariate shift in the estimated dynamics, recent methods design density estimation-based “pessimistic" penalties to prevent the learner policy from entering uncertainty regions in the state-action space (i.e., space not covered in the demonstration dataset) [250, 249, 248].

In this paper, we instead approach IRL from the Bayesian Theory of Mind perspective [105], where we *simultaneously* estimate the expert’s reward function and their *internal* model of the environment dynamics. The core idea of BTOM is that expert decisions convey their beliefs about the environment [105] and thus should affect the update direction of the dynamics model as op-

posed to it being fixed. BTOM has mostly been used to understand human biases encoded in the internal dynamics in simple and highly constrained domains [251, 109, 124, 108, 136, 129, 126]. In contrast to these works, we study how BTOM naturally enables learning high-performance and robust policies given a limited dataset.

We first propose a class of priors parameterizing how accurate we believe the expert's model of the environment is. We then show that if the expert is believed a priori to have a highly accurate model, robustness emerges naturally from BTOM's *simultaneous* estimation approach by planning against the worst-case dynamics outside the offline data distribution. We further analyze how varying the prior affects the performance of the learner agent and pair our analysis with a set of algorithms which extend prior simultaneous estimation approaches [124, 108] to high-dimensional continuous-control settings. We show that the proposed algorithms outperform state-of-the-art (SOTA) offline IRL methods without the need for designing pessimistic penalties.

In summary, our contributions are the following:

- We show that BTOM under appropriate formulation of the prior is robust to inaccuracies in the estimated dynamics model.
- We propose a set of practical algorithms for simultaneous estimation of reward and dynamics in high-dimensional environments.
- We perform extensive experiments in the MuJoCo environment to confirm our analysis and show that the proposed algorithms outperform pessimistic approaches.

6.3 Preliminaries

6.3.1 Markov Decision Process

We consider modeling agent behavior using infinite-horizon *entropy-regularized* Markov decision processes (MDP; [252]) defined by tuple $(\mathcal{S}, \mathcal{A}, \mu, P, \gamma, R)$ with state space $\mathcal{S}$, action space $\mathcal{A}$, initial state distribution $\mu(s_0) \in \Delta(\mathcal{S})$, transition probability distribution $P(s'|s,a) \in \Delta(\mathcal{S})$, discount factor $\gamma \in (0,1)$, and reward function $R(s,a) \in \mathbb{R}$. We denote the discounted occupancy

measure as $\rho_P^\pi(s,a) = \mathbb{E}_{\mu,P,\pi}\left[\sum_{t=0}^{\infty}\gamma^t P(s_t=s,a_t=a)\right]$ and the marginal state-action distribution as $d_P^\pi(s,a) = (1-\gamma)\rho_P^\pi(s,a)$. We further denote the discounted occupancy measure starting from a specific state-action pair (s,a) with $\rho_P^\pi(\tilde{s},\tilde{a}|s,a)$. The agent selects actions from an optimal policy $\pi(a|s) \in \Delta(\mathcal{A})$ that achieves the maximum expected discounted cumulative rewards and policy entropy $\mathcal{H}(\pi(a|s)) = -\sum_{\tilde{a}}\pi(\tilde{a}|s)\log\pi(\tilde{a}|s)$ in the MDP:

$$\max_\pi J(\pi) = \mathbb{E}_{\mu,P,\pi}\left[\sum_{t=0}^{\infty}\gamma^t\left(R(s_t,a_t) + \mathcal{H}(\pi(a_t|s_t))\right)\right] \tag{6.1}$$

The optimal policy satisfies the following conditions (i.e., Boltzmann rationality; [60]):

$$\begin{aligned}\pi(a|s) &\propto \exp\left(Q(s,a)\right)\\ Q(s,a) &= R(s,a) + \gamma\mathbb{E}_{P(s'|s,a)}\left[V(s')\right]\\ V(s) &= \log\sum_{a'}\exp\left(Q(s,a')\right)\end{aligned} \tag{6.2}$$

6.3.2 Inverse Reinforcement Learning

The majority of contemporary IRL approaches have converged on the Maximum Causal Entropy (MCE) IRL framework, which aims to find a reward function $R_\theta(s.a)$ with parameters θ such that the entropy-regularized learner policy $\hat{\pi}$ has matching state-action feature with the unknown expert policy π [114].

A related formulation casts IRL as maximum *discounted* likelihood (ML) estimation [253, 254, 117], subject to the constraint that the policy is entropy-regularized. Given a dataset of N expert trajectories each of length T: $\mathcal{D} = \{\tau_i\}_{i=1}^N, \tau = (s_{1:T}, a_{1:T})$ sampled from the expert policy in environment P with occupancy measure $\rho_\mathcal{D} := \rho_P^\pi$, ML-IRL aims to solve the following optimization

problem:

$$\begin{aligned} \max_{\theta} \quad & \mathbb{E}_{(s_t,a_t)\sim\mathcal{D}}\left[\sum_{t=0}^{\infty}\gamma^t \log \hat{\pi}_\theta(a_t|s_t)\right] \\ \text{s.t.} \quad & \hat{\pi}_\theta(a|s) = \arg\max_{\hat{\pi}\in\Pi} \mathbb{E}_{\rho_P^{\hat{\pi}}}\left[R_\theta(s,a) + \mathcal{H}(\hat{\pi}(\cdot|s)\right] \end{aligned} \tag{6.3}$$

where the policy is implicitly parameterized by the reward parameters θ.

It can be shown that MCE-IRL and ML-IRL are equivalent under linear reward parameterization [253, 254], however (6.3) permits non-linear reward parameterization through the following surrogate optimization problem:

$$\begin{aligned} \max_{\theta} \quad & \mathbb{E}_{\rho_\mathcal{D}}\left[R_\theta(s,a)\right] - \mathbb{E}_{\rho_P^{\hat{\pi}}}\left[R_\theta(s,a)\right] \\ \text{s.t.} \quad & \hat{\pi}_\theta(a|s) = \arg\max_{\hat{\pi}\in\Pi} \mathbb{E}_{\rho_P^{\hat{\pi}}}\left[R_\theta(s,a) + \mathcal{H}(\hat{\pi}(\cdot|s)\right] \end{aligned} \tag{6.4}$$

(6.4) can be efficiently solved via alternating training of the learner policy and the reward function, similar to Generative Adversarial Network (GAN)-based algorithms [255, 132, 256, 257, 258, 118]. However, these methods all require access to the ground truth environment dynamics or a high quality simulator in order to compute or sample from the learner occupancy measure ρ_P^π.

6.3.3 Offline Model-Based IRL & RL

Existing offline model-based IRL algorithms such as [121, 122] adapt (6.4) using a two-step process. First, an estimate $\hat{P}$ of the environment dynamics is obtained from the offline dataset, e.g., using maximum likelihood estimation. Then, $\hat{P}$ is fixed and used in place of P to compute $\rho_{\hat{P}}^\pi$ while optimizing (6.4). However, this simple replacement incurs a gap between (6.4) and (6.3) which scales with the dynamics model error and the estimated value [249]. This puts a high demand on the accuracy of the estimated dynamics.

A related challenge is to prevent the policy from exploiting inaccuracies in the estimated dynamics, which can lead to erroneously high estimated value. This has been extensively studied in both online and offline model-based RL literature [259, 49, 47, 260]. The majority of recent

offline model-based RL methods combat model-exploitation via a notion of “pessimism", which penalizes the learner policy from visiting states where the model is likely to be incorrect [259]. These pessimistic penalties are often designed based on quantifying uncertainty about transition dynamics through the estimated model [261, 262]. Drawing on these advances, recent offline IRL methods also incorporate pessimistic penalties into their RL subroutine [249, 248, 250]. However, it should be noted that designing pessimistic penalties involves nontrivial decisions to ensure that they can accurately capture out-of-distribution samples [263].

An orthogonal approach to avoid model-exploitation is to perform policy training against the worst-case dynamics in out-of-distribution states [264], similar to robust MDP [265, 266]. Rigter et al. [267] implemented this idea in the RAMBO algorithm and showed that it is competitive with pessimistic penalty-based approaches while requiring significantly less tuning. We will show that robust MDP corresponds to a sub-problem of IRL under the BTOM formulation.

6.4 Bayesian Theory of Mind

We consider IRL under the Bayesian Theory of Mind framework, where the observed expert decisions are the results of an unknown reward function $R_{\theta_1}(s, a)$ and their *internal* model of the environment dynamics $\hat{P}_{\theta_2}(s'|s, a)$. We denote the concatenated parameters with $\theta = \{\theta_1, \theta_2\}$ and condition the policy on θ as $\hat{\pi}(a|s; \theta)$ to emphasize that the expert configuration is determined by both the reward and dynamics parameters. We make no additional assumption about the expert other than that their policy is Boltzmann rational (6.2) with respect to their internal reward and dynamics. This means that their internal dynamics can potentially deviate from the true environment dynamics.

Upon observing a finite set of expert demonstrations $\mathcal{D}$, BTOM aims to compute the posterior distribution $\mathbb{P}(\theta|\mathcal{D})$ given a choice of a prior distribution $\mathbb{P}(\theta)$:

$$\begin{aligned} \mathbb{P}(\theta|\mathcal{D}) &\propto \mathbb{P}(\mathcal{D}|\theta)\mathbb{P}(\theta) \\ &= \prod_{i=1}^{N}\prod_{t=1}^{T} \hat{\pi}(a_{i,t}|s_{i,t}; \theta)\mathbb{P}(\theta) \end{aligned} \tag{6.5}$$

where we have omitted the true environment transition probabilities $\prod_{i=1}^{N}\prod_{t=1}^{T} P(s_{i,t+1}|s_{i,t},a_{i,t})$ from the likelihood because they do not depend on θ.

We consider a class of prior distributions of the form:

$$\mathbb{P}(\theta) \propto \exp\left(\lambda \sum_{i=1}^{N}\sum_{t=1}^{T} \log \hat{P}_{\theta_2}(s_{i,t+1}|s_{i,t},a_{i,t})\right) \tag{6.6}$$

where the prior precision hyperparameter λ represents how accurate we believe is the expert's model of the environment.

Let $\mathcal{L}(\theta) := \frac{1}{NT}\log \mathbb{P}(\theta|\mathcal{D})$ be the log-posterior (normalized by the data size). It can be easily verified that

$$\mathcal{L}(\theta) = \mathbb{E}_{(s,a,s')\sim\mathcal{D}}\left[\log \hat{\pi}(a|s;\theta) + \lambda \log \hat{P}_{\theta_2}(s'|s,a)\right]$$

In this paper, we consider finding a Maximum A Posteriori (MAP) estimate of the BTOM model by solving the following bi-level optimization problem:

$$\begin{aligned} \max_{\theta} \quad & \mathcal{L}(\theta) \\ \text{s.t.} \quad & \hat{\pi}(a|s;\theta) = \arg\max_{\hat{\pi}\in\Pi} \mathbb{E}_{\rho^{\hat{\pi}}_{\hat{P}}}\left[R_\theta(s,a) + \mathcal{H}(\hat{\pi}(\cdot|s))\right] \end{aligned} \tag{6.7}$$

Note that this formulation differs from (6.3) and the decoupled approaches because it includes log likelihood of the dynamics in the objective (weighted by λ).

It should be noted that obtaining the full posterior distribution (or an approximation) is feasible using popular approximate inference methods (e.g., stochastic variational inference or Langevin dynamics; [268, 269]) and does not significantly alter the proposed estimation principles and algorithms.

6.4.1 Naive Solution

We start by presenting a naive solution to (6.7) which can be seen as an extension of the tabular simultaneous reward-dynamics estimation algorithms proposed by Herman et al. [124] and Wu

et al. [108] to the high-dimensional setting.

Solving (6.7) requires: 1) computing the optimal policy with respect to θ, and 2) computing the gradient $\nabla_\theta \log \hat{\pi}(a|s;\theta)$ which requires inverting the policy optimization process itself. Both operations can be done exactly in the tabular setting as in prior works but are intractable in high-dimensional settings. We propose to overcome the intractability using sample-based approximation.

In this section, we focus on approximating the gradient of the policy $\nabla_\theta \log \hat{\pi}(a|s;\theta)$, which is less obvious. We can show that the $\nabla_\theta \log \hat{\pi}(a|s;\theta)$ has the following form (see Appendix D.1 for all proofs and derivations):

$$\begin{aligned} \nabla_\theta \log \hat{\pi}(a|s;\theta) &= \nabla_\theta Q_\theta(s,a) - \nabla_\theta V_\theta(s) \\ &= \nabla_\theta Q_\theta(s,a) - \mathbb{E}_{\tilde{a}\sim\hat{\pi}(\cdot|s;\theta)}[\nabla_\theta Q_\theta(s,\tilde{a})] \end{aligned} \tag{6.8}$$

where $\nabla_\theta Q_\theta(s,a) = [\nabla_{\theta_1} Q_\theta(s,a), \nabla_{\theta_2} Q_\theta(s,a)]$ is the concatenation of reward and dynamics gradients defined as:

$$\nabla_{\theta_1} Q_\theta(s,a) = \mathbb{E}_{\rho^{\hat{\pi}}_{\hat{P}}(\tilde{s},\tilde{a}|s,a)}\left[\nabla_{\theta_1} R_{\theta_1}(\tilde{s},\tilde{a})\right] \tag{6.9}$$

$$\nabla_{\theta_2} Q_\theta(s,a) = \mathbb{E}_{\rho^{\hat{\pi}}_{\hat{P}}(\tilde{s},\tilde{a}|s,a)}\left[\gamma \sum_{s'} V_\theta(s') \nabla_{\theta_2} \hat{P}_{\theta_2}(s'|\tilde{s},\tilde{a})\right] \tag{6.10}$$

Given (6.9) and (6.10) are tractable to compute using sample-based approximation of expectations, we construct the following surrogate objective $\tilde{L}(\theta)$ with the same gradient as the original MAP estimation problem (6.7):

$$\tilde{\mathcal{L}}(\theta) = \mathbb{E}_{(s,a)\sim\mathcal{D}}[\mathcal{E}_\theta(s,a)] - \mathbb{E}_{s\sim\mathcal{D},a\sim\hat{\pi}}[\mathcal{E}_\theta(s,a)] + \lambda \mathbb{E}_{(s,a,s')\sim\mathcal{D}}[\log \hat{P}_{\theta_2}(s'|s,a)] \tag{6.11}$$

where

$$\mathcal{E}_\theta(s,a) = \mathbb{E}_{\rho^{\hat{\pi}}_{\hat{P}}(\tilde{s},\tilde{a}|s,a)}\left[R_\theta(\tilde{s},\tilde{a}) + \gamma EV_\theta(\tilde{s},\tilde{a})\right] \tag{6.12}$$

$$EV_\theta(s,a) = \sum_{s'} \hat{P}_{\theta_2}(s'|s,a)V_\theta(s') \tag{6.13}$$

Optimizing (6.11) is now the same as optimizing (6.7) but tractable.

An interesting consequence of maximizing the first two terms of (6.11) alone (excluding the prior) is that we both increase the reward and modify the internal dynamics to generate states with higher expected value (EV) upon taking expert actions then following the learner policy $\hat{\pi}$, and we do the opposite when taking learner actions. Intuitively, reward and dynamics play complementary roles in determining the value of actions and thus should be regularized [270, 109, 131]. Otherwise, one cannot disentangle the effect of truly high reward and falsely optimistic dynamics. Our prior (6.6) alleviates this unidentifiability to some extent.

6.4.2 A Robust BTOM Model

We now present our main observation that the IRL learner exhibits robust performance as a natural consequence of the BTOM formulation under the dynamics accuracy prior (6.6).

We start by analyzing a discounted, full-trajectory version of the BTOM likelihood (6.7). Note that discounting does not change the optimal solution to (6.7) under expressive reward and dynamics model class; nor does it require infinite data because we can truncate the summation at $T = \text{int}\left(\frac{1}{1-\gamma}\right)$ and obtain nearly the same estimator as with infinite sequence length. We restate a

decomposition of the discounted likelihood in [249] as follows:

$$
\begin{aligned}
&\mathbb{E}_{P(\tau)}\left[\sum_{t=0}^{\infty}\gamma^t \log \hat{\pi}_\theta(a_t|s_t)\right] \\
&= \mathbb{E}_{P(\tau)}\left[\sum_{t=0}^{\infty}\gamma^t \left(Q_\theta(s_t,a_t) - V_\theta(s_t)\right)\right] \\
&= \mathbb{E}_{\rho^\pi_P}\Big[R_{\theta_1}(s_t,a_t) + \gamma \mathbb{E}_{s'\sim\hat{P}}[V_\theta(s')]\Big] - \mathbb{E}_{\rho^\pi_P}\Big[V_\theta(s_t)\Big] \\
&= \underbrace{\mathbb{E}_{\rho^\pi_P}\Big[R_{\theta_1}(s_t,a_t)\Big] - \mathbb{E}_{\mu}\Big[V_\theta(s_0)\Big]}_{\ell(\theta)} + \underbrace{\gamma \mathbb{E}_{\rho^\pi_P}\Big[\mathbb{E}_{s'\sim\hat{P}(\cdot|s_t,a_t)}V_\theta(s') - \mathbb{E}_{s''\sim P(\cdot|s_t,a_t)}V_\theta(s'')\Big]}_{\textbf{T1}}
\end{aligned}
\tag{6.14}
$$

where **T1** corresponds to the value difference under the real and estimated dynamics. We can show that **T1** is negligible if the estimated dynamics is accurate under the *expert* data distribution:

Lemma 1. *Let* $\epsilon = \mathbb{E}_{(s,a)\sim P(\tau)} D_{KL}(P(\cdot|s,a)||\hat{P}(\cdot|s,a))$ *be the dynamics estimation error and* $R_{max} = \max_{s,a}|R_\theta(s,a)| + \log|\mathcal{A}|$ *be an upper bound on reward and policy entropy, it holds that*

$$
|\textbf{\textit{T1}}| \leq \frac{\gamma R_{max}}{(1-\gamma)^2}\sqrt{2\epsilon} \tag{6.15}
$$

Thus, if $\mathbb{E}_{(s,a)\sim P(\tau)} D_{KL}(P(\cdot|s,a)||\hat{P}(\cdot|s,a)) \leq \epsilon$ holds for sufficiently small ϵ, for example by setting a large λ, **T1** can be dropped from (6.14) and the discounted likelihood reduces to $\ell(\theta)$.

$\ell(\theta)$ highlights the reason why the proposed BTOM approach can be robust to a limited dataset. It poses the offline IRL problem as maximizing the cumulative reward of expert trajectories in the real environment, and minimizing the cumulative reward generated by the learner in the estimated dynamics with respect to *both* reward and dynamics. In other words, it aims to find performance-matching reward and policy under the *worst-case, pessimistic* dynamics, which is trained adversarially outside the data distribution. This connects BTOM to the robust MDP approach to offline model-based RL [264, 267].

Algorithm 2 Deep Bayesian Theory of Mind (BTOM)

Require: Dataset $\mathcal{D} = \{\tau\}$, dynamics model $\hat{P}_{\theta_2}(s'|s,a)$, reward model $R_{\theta_1}(s,a)$, hyperparameters λ_1, λ_2

1: **for** $k = 1 : K$ **do**
2: Run MBPO to update learner policy $\hat{\pi}(a|s;\theta)$ and value function $Q_\theta(s,a)$ in dynamics $\hat{P}$
3: Sample real trajectory τ_{real} starting from $(s,a) \sim \mathcal{D}$ and following $\hat{P}$ and $\hat{\pi}$
4: Sample fake trajectory τ_{fake} starting from $s \sim \mathcal{D}, a_{\text{fake}} \sim \hat{\pi}(\cdot|s;\theta)$ and following $\hat{P}$ and $\hat{\pi}$
5: Evaluate (6.16) and take a gradient step
6: Evaluate (6.17) and take a few gradient steps.
7: **end for**

6.4.3 Proposed Algorithms

Using the insights from the previous sections, we propose two scalable Deep Bayesian Theory of Mind algorithms to find the MAP solution to (6.7). The first algorithm (**BTOM**; 2) applies the naive solution with surrogate objective (6.11), while the second algorithm (**RTOM**; 3) exploits the observation in section 6.4.2 to derive a more efficient algorithm for high λ via surrogate objective $\ell(\theta)$.

The estimation problem (6.7) has an inherently nested structure where, for each update of parameters θ (the outer problem), we have to solve for the optimal policy $\hat{\pi}(a|s;\theta)$ (the inner problem). Following recent ML-IRL approaches [254, 249], we perform the nested optimization using *two-timescale* stochastic approximation [271, 272], where the inner problem is solved via stochastic gradient updates on a faster time scale than the outer problem. For both algorithms, we solve the inner problem using Model-Based Policy Optimization (MBPO; [47]) which uses Soft Actor-Critic (SAC; [60]) in a dynamics model ensemble.

BTOM. For the BTOM outer problem, we estimate the expectations in (6.11) and (6.12) via sampling and perform coordinate-ascent optimization. Specifically, for each update step, we first sample a mini-batch of state-action pairs $(s,a) \sim \mathcal{D}$ and a mini-batch of (fake) actions $a_{\text{fake}} \sim \hat{\pi}(\cdot|s;\theta)$ and simulate both (s,a) and (s,a_{fake}) forward in the estimated dynamics $\hat{P}$ to get the real and fake trajectories $\tau_{\text{real}}, \tau_{\text{fake}}$. We then optimize the reward function first by taking a single

Algorithm 3 Robust Theory of Mind (RTOM)

Require: Dataset $\mathcal{D} = \{\tau\}$, dynamics model $\hat{P}_{\theta_2}(s'|s,a)$, reward model $R_{\theta_1}(s,a)$, hyperparameters λ_1, λ_2

1: **for** $k = 1 : K$ **do**
2: Run MBPO to update learner policy $\hat{\pi}(a|s;\theta)$ and value function $Q_\theta(s,a)$ in dynamics $\hat{P}$
3: Sample fake trajectory τ_{fake} starting from $s \sim \mathcal{D}$ and following $\hat{P}$ and $\hat{\pi}$
4: Evaluate (6.19) and take a gradient step
5: Evaluate (6.20) and take a few gradient steps
6: **end for**

gradient step to optimize the following objective function:

$$\max_{\theta_1} \quad \mathbb{E}_{(s,a)\sim\mathcal{D},\rho^{\hat{\pi}}_{\hat{P}}(\tilde{s},\tilde{a}|s,a)}\left[R_{\theta_1}(\tilde{s},\tilde{a})\right] - \mathbb{E}_{s\sim\mathcal{D},a_{\text{fake}}\sim\hat{\pi},\rho^{\hat{\pi}}_{\hat{P}}(\tilde{s},\tilde{a}|s,a_{\text{fake}})}\left[R_{\theta_1}(\tilde{s},\tilde{a})\right] \tag{6.16}$$

Lastly, we optimize the dynamics model by taking a few gradient steps (a hyperparameter) to optimize the following objective function using on-policy rollouts branched from mini-batches of expert state-actions as in RAMBO [267]:

$$\begin{aligned}\max_{\theta_2} \quad & \lambda_1\mathbb{E}_{(s,a)\sim\mathcal{D},\rho^{\hat{\pi}}_{\hat{P}}(\tilde{s},\tilde{a}|s,a)}\left[EV_{\theta_2}(\tilde{s},\tilde{a})\right] - \lambda_1\mathbb{E}_{s\sim\mathcal{D},a_{\text{fake}}\sim\hat{\pi},\rho^{\hat{\pi}}_{\hat{P}}(\tilde{s},\tilde{a}|s,a_{\text{fake}})}\left[EV_{\theta_2}(\tilde{s},\tilde{a})\right] \\ & + \lambda_2\mathbb{E}_{(s,a,s')\sim\mathcal{D}}\left[\log\hat{P}_{\theta_2}(s'|s,a)\right]\end{aligned} \tag{6.17}$$

We estimate the dynamics gradient using the REINFORCE method with baseline:

$$\begin{aligned}\nabla_{\theta_2}EV_\theta(s,a) &= \sum_{s'}V_\theta(s')\nabla_{\theta_2}\hat{P}_{\theta_2}(s'|s,a) \\ &= \mathbb{E}_{s'\sim\hat{P}(\cdot|s,a)}\left[\left(V_\theta(s') - b(s,a)\right)\nabla_{\theta_2}\log\hat{P}_{\theta_2}(s'|s,a)\right]\end{aligned} \tag{6.18}$$

Following Rigter et al. [267], we set the baseline to $b(s,a) = Q_\theta(s,a) - R_{\theta_1}(s,a)$ to reduce gradient variance and further normalize $V_\theta(s') - b(s,a)$ across the mini-batch to stabilize training. In the continuous-control setting, the value function can be estimated as $V_\theta(s) = \mathbb{E}_{a\sim\hat{\pi}_\theta}[Q_\theta(s,a) - \log\hat{\pi}(a|s;\theta)]$ with a single sample.

RTOM. We adapt the BTOM algorithm slightly for the RTOM outer problem, where we only simulate a single trajectory for each state in the mini-batch and update the reward using the following objective:

$$\max_{\theta_1} \mathbb{E}_{\rho_{\mathcal{D}}} \left[R_{\theta_1}(s,a) \right] - \mathbb{E}_{\rho^{\hat{\pi}}_{\hat{P}}} \left[R_{\theta_1}(s,a) \right] \tag{6.19}$$

We then update the dynamics by dropping the first term in (6.17):

$$\max_{\theta_2} \quad - \lambda_1 \mathbb{E}_{s\sim\mathcal{D}, a_{\text{fake}}\sim\hat{\pi}, \rho^{\hat{\pi}}_{\hat{P}}(\tilde{s},\tilde{a}|s,a_{\text{fake}})} \left[EV_{\theta_2}(\tilde{s},\tilde{a}) \right] + \lambda_2 \mathbb{E}_{(s,a,s')\sim\mathcal{D}} \left[\log \hat{P}_{\theta_2}(s'|s,a) \right] \tag{6.20}$$

We provide additional details about the proposed algorithms in Appendix D.2.

6.4.4 Performance Guarantees

In this section, we study how policy and dynamics estimation error affect learner performance in the real environment. Vemula et al. [273] provided the following result relating expert-learner performance gap in the real and estimated environment in the context of model-based RL:

Lemma 2. *(Performance difference via advantage in model; Lemma 4.1 in [273]) Let d^{π}_{P} denote the marginal state-action distribution following policy π in environment P. The following relationship holds:*

$$\mathbb{E}_{(s,a)\sim d^{\pi}_{P}} \left[\log \hat{\pi}_{\hat{P}}(a|s) \right] = \mathbb{E}_{s\sim d^{\pi}_{P}} \left[\mathbb{E}_{a\sim\pi} Q^{\hat{\pi}}_{\hat{P}}(s,a) - V^{\hat{\pi}}_{\hat{P}}(s) \right] \tag{6.21}$$

$$= \underbrace{(1-\gamma)\mathbb{E}_{s\sim\mu} \left[V^{\pi}_{P}(s) - V^{\hat{\pi}}_{P}(s) \right]}_{\textit{Performance difference in real environment}} \tag{6.22}$$

$$+ \underbrace{\gamma \mathbb{E}_{(s,a)\sim d^{\hat{\pi}}_{P}} \left[\mathbb{E}_{s'\sim P} V^{\hat{\pi}}_{\hat{P}}(s') - \mathbb{E}_{s''\sim\hat{P}} V^{\hat{\pi}}_{\hat{P}}(s'') \right]}_{\textit{Model (dis)advantage under learner distribution}} \tag{6.23}$$

$$+ \underbrace{\gamma \mathbb{E}_{(s,a)\sim d^{\pi}_{P}} \left[\mathbb{E}_{s'\sim\hat{P}} V^{\hat{\pi}}_{\hat{P}}(s') - \mathbb{E}_{s''\sim P} V^{\hat{\pi}}_{\hat{P}}(s'') \right]}_{\textit{Model advantage under expert distribution}} \tag{6.24}$$

Intuitively, maximizing the policy likelihood (6.21) w.r.t. $\hat{P}$ (including the reward) increases

the performance gap (6.22) between the expert and the learner, increases model advantage under the expert data distribution, and decreases model advantage under the (unknown) learner data distribution. The performance gap is then to be closed by the learner during the inner optimization problem.

Using this result, we arrive at the follow performance bound:

Theorem 3. *Let* $\epsilon_{\hat{P}} = \mathbb{E}_{(s,a)\sim d_P^{\pi}} D_{KL}[P(\cdot|s,a)||\hat{P}(\cdot|s,a)]$ *be the dynamics estimation error and* $\epsilon_{\hat{\pi}} = -\mathbb{E}_{(s,a)\sim d_P^{\pi}}[\log \hat{\pi}_{\hat{P}}(a|s)]$ *be the policy estimation error. Assuming bounded expert-learner marginal state-action density ratio* $\left\| \frac{d_P^{\hat{\pi}}(s,a)}{d_P^{\pi}(s,a)} \right\|_{\infty} \leq C$*, we have the following (absolute) performance bound for the IRL agent:*

$$|J_P(\hat{\pi}) - J_P(\pi)| \leq \frac{1}{1-\gamma}\epsilon_{\hat{\pi}} + \frac{\gamma(C+1)R_{max}}{(1-\gamma)^2}\sqrt{2\epsilon_{\hat{P}}} \tag{6.25}$$

This bound highlights the connection between IRL and behavior cloning and the Bayesian nature of IRL: by incorporating the dynamics and Bellman-optimality as regularizations, we can achieve better generalizations than behavior cloning. We believe a tighter bound can be obtained by further analyzing the density ratio C given that the BTOM policy will act conservatively as a result of planning against worst-case dynamics. We leave this to future work.

6.5 Experiments

We aim to answer the following questions with our experiments:

1. How does the dynamics accuracy prior affect BTOM agent behavior?

2. How well does BTOM and RTOM perform compared to SOTA offline IRL algorithms?

We investigate Q1 using a Gridworld environment. We investigate Q2 using the standard D4RL dataset on MuJoCo continuous control benchmarks.

6.5.1 Gridworld Example

We use a 5x5 gridworld environment to understand the behavior of the BTOM algorithm. The environment has deterministic transitions conditioned on the following set of actions: up, down,

left, right, and stay. Any actions pointing in the direction of the boundary when the agent is already in a boundary cell will keep the agent in the same cell. The expert agent, who knows the true transition dynamics and plans using a discount factor of $\gamma = 0.7$, starts in the lower left corner and receives a reward when reaching the upper right corner. We represent the reward function as the log probability of the target state: $\log \tilde{P}(s)$, where the upper right corner has a target probability of 1.

Using 100 expert trajectories of length 50, we trained 3 BTOM agents with transition likelihood penalty λ of 0.001, 0.5, and 10, respectively. As a comparison, we also trained a decoupled agent whose dynamics model is fixed after an initial maximum likelihood pretraining step and its reward is estimated using the same gradient update rule as BTOM in (6.9).

Given that the environment is simple and both the policy, reward, and dynamics models are well-specified, all agents recover the ground-truth policy in state-actions pairs visited by the expert. The ground truth and estimated target state probabilities are shown in the first row of Fig. 6.1. All agents correctly estimated that the upper right corner has the highest reward, although not with the same precision as the ground truth sparse reward. BTOM agents with $\lambda = 0.5$ and $\lambda = 10$ are able to assign high reward only to states close to the true goal state, where as the BTOM agent with $\lambda = 0.001$ and the decoupled agent assigned high rewards to state much further away from the true goal state.

Table 6.1: MuJoCo benchmark performance using 10 expert trajectories from the D4RL dataset. Each row reports the mean and standard deviation of performance over 5 random seeds.

Environment	Dataset	BTOM (ours)	RTOM (ours)	ML-IRL	Expert
HalfCheetah	Medium	$\mathbf{8813.35 \pm 997.49}$	8085.18 ± 597.86	7706.43 ± 159.39	12156.16 ± 88.01
HalfCheetah	Medium-replay	7508.65 ± 190.75	6961.28 ± 130.61	$\mathbf{9383.34 \pm 358.67}$	12156.16 ± 88.01
HalfCheetah	Medium-expert	$\mathbf{11519.98 \pm 149.69}$	11289.09 ± 258.70	11276.09 ± 551.94	12156.16 ± 88.01
Hopper	Medium	2243.15 ± 922.75	$\mathbf{3306.59 \pm 473.60}$	2461.45 ± 705.70	3512.64 ± 17.10
Hopper	Medium-replay	$\mathbf{3520.69 \pm 29.50}$	3307.11 ± 471.38	2889.73 ± 542.65	3512.64 ± 17.10
Hopper	Medium-expert	3209.91 ± 731.66	$\mathbf{3550.25 \pm 28.85}$	3350.79 ± 264.96	3512.64 ± 17.10
Walker2D	Medium	$\mathbf{4307.99 \pm 855.55}$	4035.21 ± 247.23	4195.36 ± 352.86	5365.62 ± 55.79
Walker2D	Medium-replay	3960.70 ± 1521.52	3880.54 ± 713.29	$\mathbf{4092.58 \pm 308.71}$	5365.62 ± 55.79
Walker2D	Medium-expert	4862.66 ± 100.37	$\mathbf{4941.10 \pm 38.99}$	4363.54 ± 729.60	5365.62 ± 55.79

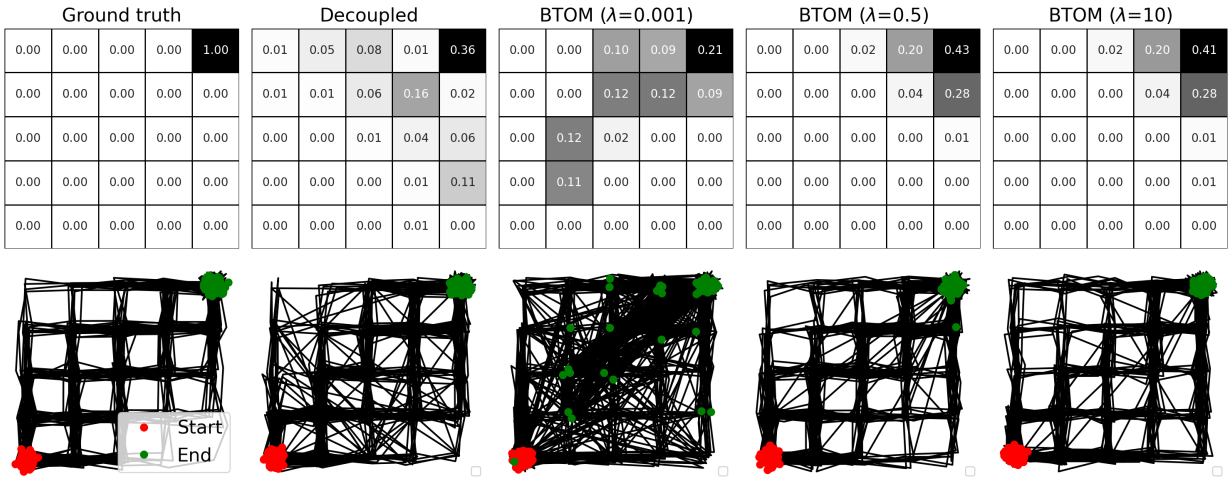


Figure 6.1: Gridworld experiment results. **(Row 1)** Ground truth and estimated target state distributions (softmax of reward) for agents using decoupled estimation and BTOM agents with $\lambda = [0.001, 0.5, 10]$. BTOM agents with higher λ obtain more accurate reward estimates. **(Row 2)** Sample paths generated by the ground truth agent, decoupled, and BTOM agents. BTOM agents with higher λ generate fewer illegal (diagonal) transitions. Illegal transitions generated by BTOM agents have a strong tendency to point towards the goal state.

We visualize the estimated dynamics models by sampling 100 imagined rollouts using the estimated policies in the second row of Fig. 6.1. This figure shows that the BTOM($\lambda = 0.001$) and the decoupled agent would take significantly more illegal transitions (i.e., diagonal transitions) than BTOM agents with higher λ. Comparing among BTOM agents, we see that increasing λ decreases the number of illegal transitions. In contrast to the decoupled agent whose illegal transitions are rather random, the illegal transitions generated by BTOM agents with lower λ have a strong tendency to point towards the goal state. This corroborates with our analysis that BTOM optimizes model advantage under the expert distribution.

6.5.2 MuJoCo Benchmarks

In this section, we compare the performance of BTOM and RTOM with SOTA offline IRL algorithms in the MuJoCo continuous control environments [274] using the D4RL dataset [275]. We use ML-IRL [249], an offline model-based IRL algorithm based on MOPO [261], as our comparison.

We use the following MuJoCo environments: HalfCheetah, Hopper, and Walker2D. For each environment, D4RL offers 4 types of datasets: medium, medium-replay, medium-expert, and expert. Following prior IRL evaluation protocols, our agents maintain two datasets: 1) a *transition dataset* is used to train the dynamics model and the actor-critic networks and 2) an *expert dataset* is used to train the reward function. The transition dataset is selected from one of the first three types of D4RL datasets and is not sub-sampled. The expert dataset contains 10 randomly sampled D4RL expert trajectories. For both BTOM and RTOM, we set the model objective weighting terms to $\lambda_1 = 0.01, \lambda_2 = 1$ to encourage an accurate model under the data distribution. For each environment and transition dataset, we train our algorithms for a fixed number of epochs and repeat this process for 5 random seeds. After the final epoch, we evaluate the agent for 10 episodes in the MuJoCo environments. We provide additional implementation and hyperparameter details in Appendix D.2.

Table 6.1 reports the mean and standard deviation of the evaluation performance across different seeds for each setting. For ML-IRL, we list the results reported in the original publication. Our algorithms outperform the benchmark in almost all settings. On the medium-expert dataset, which has the best coverage of expert trajectories, our algorithms perform near optimally and overall have smaller variance than ML-IRL.

Between the two proposed algorithms, BTOM and RTOM perform comparably on the medium-expert datasets. However, BTOM outperforms RTOM on the medium and medium-replay datasets in the Halfcheetah and Walker2D environments. Training the dynamics model on these datasets corresponds to violating the dynamics accuracy assumption for optimizing only $\ell(\theta)$ in (6.14) as **T1** would be large in this case. For BTOM, this is not a problem because the dynamics log likelihood only serves as a prior and the surrogate objective (6.11) is not affected. However, for RTOM, relaxing the dynamics accuracy assumption causes $\ell(\theta)$ to deviate from the true objective.

Finally, we remark that BTOM has less stable training dynamics than RTOM where its evaluation performance may alternate between periods of near optimal performance and periods of medium performance (thus the larger variance in Table 6.1). While stability is a known issue for

training energy-based models using contrastive divergence objectives (i.e., objective (6.11); [276]), we believe the current issue is related to BTOM's two-sample path method having weaker and noisier learning signal. Another source of instability is likely introduced by simultaneously training the dynamics model, which may be improved in future work by adding Lipschitz regularizations [277].

6.6 Related Work and Discussions

Bayesian IRL. Ramachandran and Amir [113] first proposed a Bayesian formulation of IRL to solve the reward ambiguity problem. A MAP inference approach was proposed in [278] and a variational inference approach was proposed in [245]. Their formulations consider non-entropy-regularized policies and the dynamics model is fixed during reward inference. In contrast, simultaneous estimation of reward and dynamics can potentially infer the demonstrator's biased beliefs about the environment, which is desirable for psychology and human-robot interaction studies [105, 108, 109]. Despite the attractiveness, simultaneous estimation is challenging because of the need to invert the agent's planning process, especially in continuous domains. Reddy et al. [109] avoids this by representing agent discrete choice policies using neural network-parameterized Q functions and regularizing the Bellman error to be small over the entire state-action space. This method however cannot be straightforwardly adapted to the continuous action case. Kwon et al. [128] avoids this by first training a task-conditioned policy on a distribution of environments with known parameters using meta reinforcement learning and then use the meta-trained policy to guide inference. This precludes the method from being used in general settings with unknown task distributions. To our knowledge, our proposed algorithms are the first to address simultaneous estimation in general environments.

Decision-aware model learning. Decision-aware model learning aims to solve the objective mismatch problem in model-based RL [46]. Many proposed methods in this class use value-targeted regression similar to our model loss in (6.17) [53, 279]. Our analysis and that of Vemula et al. [273] suggest that value-targeted model objectives may be related to robust objectives. Furthermore, since the set of value-equivalent models only shrink for an increasingly larger set of

policy and values [53], using value-aware model objectives alone may not be optimal and additional prediction-based regularizations may be needed.

Theory of Mind. Theory of Mind inference is known to be unidentifiable in general. Many researchers believe that reliable inference in human theory of mind relies on highly structured priors and normative assumptions [14, 270, 280]. We took a small step in understanding the relationship between a type of structured prior, i.e., the dynamics accuracy prior (6.6), and the inference outcome. Different from prior works which also use accuracy-based regularizations but assume known ground truth dynamics [109, 131], our prior is more general and flexible since it is estimated partially from data. While our goal in this work has been to understand BTOM inference of expert demonstrators, an interesting future direction is to identify appropriate priors to reliably infer reward and internal dynamics from sub-optimal and biased human demonstrators.

Our observation of the robustness of BTOM also has interesting cognitive science implications. It suggests that inference of (Boltzmann) rational agents naturally gives rise to a form of "pessimism in the face of uncertainty", which provides a testable hypothesis of Boltzmann rationality as a model of human theory of mind. Furthermore, this knowledge can potentially be applied in machine teaching and multi-agent coordination settings to design more efficient and human-like communicative actions [281, 282, 283].

6.7 Conclusion

We showed that inverse reinforcement learning under the Bayesian Theory of Mind framework gives rise to robust policies. This yielded a set of novel offline model-based IRL algorithms achieving SOTA performance in the MuJoCo continuous control benchmarks without ad hoc pessimistic penalty design.

7. CONCLUSIONS

This dissertation serves as an attempt to understand the role of beliefs and cognitive dynamics in human decision making. The main idea pursued is that *belief-understanding is as important as desire-understanding, if not more*. In particular, I focused on the Theory of Mind framework to infer unobservable cognitive dynamics from observable human behavior. TOM inference requires a rationalist model of decision making which makes proper use of beliefs to realize desire in uncertain environments. While there is no consensus on how to optimally leverage beliefs for actions in an efficient and human-like way, at least not in the current state of reinforcement learning and optimal control, active inference can serve as a promising framework for thinking about the enactive role of beliefs.

In Chapter 2, I provided a throughout review of active inference and contrasted it with established alternative decision-making paradigms, namely, reinforcement learning and optimal control. As much as active inference attempts to unify belief and desire, its plagued by the hardness of forward design of agent objectives. I thus focused instead on reverse-engineering agent objectives via TOM. Chapter 3 highlighted an important shortcoming in existing computational models of TOM – the degeneracy of joint belief-desire inference. While this degeneracy has been acknowledged before [280], it's implications and mitigation strategies have rarely been studied in the literature. My observations highlight that realistic TOM requires complex priors that are grounded in the actual environment and tasks as opposed to an arbitrary large hypothesis space.

Using these insights, I conducted a series of experiments to illustrate that joint belief-desire inference has the benefit of allowing us to:

1. Rationalize seemingly sub-optimal human behavior in abnormal scenarios

2. Inspect and correct model failures and obtain robust control policies without ad-hoc design or experimentation

Specifically, in Chapter 4, I proposed an active inference-factor analysis framework to under-

stand variations in human behavior as differences in their mental models (i.e., beliefs) of the environment. Applying this framework to the analysis of driver responses to automated vehicle failures, I showed that variations in driver emergency braking reactions can be understood as different levels of trust and situation awareness, which are encoded as the principal components of the factor model. This framework can potentially complement the currently dominant hypothesis-driven approach to the study of human trust and situation awareness in human-automation interaction. The advantage provided by the proposed framework is the ability to elicit human cognitive behavior without intrusive or subjective measurements in realistic environments.

In Chapter 5, I investigated the advantages of belief modeling by comparing an active inference model scaled up using artificial neural networks to established rule-based and black-box neural network driver models in a car-following task. I showed that active inference outperforms neural network models which are known to suffer from covariate shift in the limited data setting, and it generalizes better than rule-based models thanks to its flexible structure. More importantly, the modularity of the active inference model affords a natural interpretability mechanism via visualizations of model beliefs, which helps designers fully comprehend how model outputs are produced from inputs. This enables designers to efficiently inspect and correct model failures, which in the present case are caused by limited training data. Being able to quickly comprehend model behavior and make precise editing decisions without extensive queries or experimentation contributes to more transparent and reliable models demanded by modern ML systems [195].

In Chapter 6, I investigated the root of belief modeling's superior performance as observed in Chapter 5. The analysis showed that the superiority is primarily due to a better handling of uncertainty in the low data regions of the state space through a robust formulation. The robust formulation is a natural consequence of belief modeling under a special family of accuracy-promoting priors. Importantly, this means that learning high-performance control policies from expert demonstrations needs no ad-hoc modifications, such as designing pessimistic penalties. The performance advantage was clearly demonstrated through a benchmark comparison against state of the art offline inverse reinforcement learning algorithms on high-dimensional continuous control tasks. Fur-

thermore, the analysis in this chapter highlighted the Bayesian nature of TOM: by incorporating sequential decision making structure as a prior on the hypothesis class of demonstrator policies, we can achieve a smaller performance gap and better generalization than supervised learning.

Overall, these results have demonstrated the central role of belief modeling in understanding human behavior and its advantages in engineering transparent, reliable, and capable artificial agents.

7.1 Future Directions

Inverse reinforcement learning. The field of reinforcement learning and optimal control has progressed significantly in the past few years (two years to be more precise) with the proposal of incorporating novel model architectures (such as Transformers and diffusion models; [284, 285]) and ideas to circumvent the cumbersome dynamic programming problem (e.g., using RL-via-supervised-learning; [286]). In contrast, inverse reinforcement learning, and the related field of imitation learning, has been rather stagnant, where most of the recent ideas are minor modifications of maximum entropy IRL and related adversarial IRL frameworks [115, 132, 118]. Part of the reason is that IRL is has mostly been treated as a sub-field of RL where the only differentiator is the reward learning loop on top of regular RL. This thesis aims to advocate for a broader view of IRL, namely by framing it as a more general problem of Bayesian inference.

A central challenge of IRL is to solve a difficult planning problem in the inner loop and then invert the planning process in the outer loop. Most IRL approaches, including the ones presented in this thesis, follow this recipe and solve the RL problem using dynamic programming or temporal difference learning. Under this paradigm, the most straightforward way to enhance current IRL methods is to use more efficient planning methods for the problem of interest (e.g., [287]), or more efficient ways to invert the planning problem (e.g., using implicit differentiation [50] or other ways to bypass the bi-level optimization problem [288]). However, the true power lies in not framing IRL as inverse planning. The discussions in Chapter 3 and 6 highlighted the benefits of framing IRL as an inference problem and taking advantage of flexible priors. Following this idea, one way to overcome the inverse planning problem is to formulate rationality as priors or constraints. This has

already been explored by Reddy et al. [109], Chan and van der Schaar [245], and Piot et al. [289], but has not been applied to large-scale problems. Furthermore, specifying rationality as priors allows us to model other types of rationality or sub-optimality [290], where ignorance or mis-specification of rationality can significantly reduce inference accuracy [291]. How to design these priors such that the inferred reward is not degenerate and transferable is an interesting problem.

Theory of mind. Practical theory of mind inference requires two ingredients: accuracy and efficiency. While I have extensively studied the inaccuracy/unidentifiability of naive TOM (Chapter 3) and proposed a set of solutions to overcome its deficiencies from a performance-driven perspective, they do not directly address how to reliably obtain biased beliefs, which is of higher interest in real applications. As explained by Armstrong and Mindermann [230], eliciting biased beliefs requires highly complex priors. Rabinowitz et al. [292] used meta-learning to automatically construct the prior from synthetic data. However, the adopted meta-learning approach requires expert design of the meta-training data and lacks interpretability.

To equip actual ML systems with TOM abilities, the inference algorithms also need to be fast just as much as they need to be accurate. The algorithms proposed in this dissertation do not satisfy this criteria, since each algorithm takes a few hours to run. Hadfield-Menell et al. [293] proposed to reduce inference time by distilling the desired behavior of a TOM agent in a neural network via reinforcement learning. While their approach eschewed the possibility of explicit reasoning about belief and desire, which is crucial to interpretability, their work suggested a promising direction of using learned/amortized optimization to speed up time-consuming inference, much like variational inference and meta-learning [294].

There is thus exciting work to be done in combining Bayesian inference and meta-learning for both building and eliciting the knowledge of trained agents and speeding up test-time inference, especially given the close connection between the two paradigms [295, 296]. For examples, the experiments by Rabinowitz et al. [292] have already implied that TOM most likely emerge with well-structured variety of environments and tasks and agents with different levels of rationalities. Mikulik et al. [297] presented an approach to elicit the beliefs of meta-trained agents by fitting

surrogate models. There is also an increasing amount of work in online variational inference and learning [298]. Carefully combining these insights will likely lead to more progress on transparent, reliable, and value-aligned agents.

REFERENCES

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[4] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[5] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

[6] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017.

[7] Joshua S Rule, Joshua B Tenenbaum, and Steven T Piantadosi. The child as hacker. *Trends in cognitive sciences*, 24(11):900–915, 2020.

[8] Barry Brown. The social life of autonomous cars. *Computer*, 50(2):92–96, 2017.

[9] Hananeh Alambeigi, Anthony D McDonald, and Srinivas R Tankasala. Crash themes in automated vehicles: A topic modeling analysis of the california department of motor vehicles automated vehicle crash database. *Transportation Research Board, 99th Annual Meeting, Washington D.C.*, 2020.

[10] Alison Gopnik and Janet W Astington. Children's understanding of representational change and its relation to the understanding of false belief and the appearance-reality distinction.

Child development, pages 26–37, 1988.

[11] Chris L Baker, Julian Jara-Ettinger, Rebecca Saxe, and Joshua B Tenenbaum. Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. *Nature Human Behaviour*, 1(4):1–10, 2017.

[12] Michael Rescorla. Bayesian modeling of the mind: From norms to neurons. *Wiley Interdisciplinary Reviews: Cognitive Science*, 12(1):e1540, 2021.

[13] Gerd Gigerenzer and Wolfgang Gaissmaier. Heuristic decision making. *Annual review of psychology*, 62:451–482, 2011.

[14] Julian Jara-Ettinger. Theory of mind as inverse reinforcement learning. *Current Opinion in Behavioral Sciences*, 29:105–110, 2019.

[15] Ran Wei, Anthony D McDonald, Alfredo Garcia, and Hananeh Alambeigi. Modeling driver responses to automation failures with active inference. *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[16] Ran Wei, Anthony D McDonald, Alfredo Garcia, Gustav Markkula, Johan Engstrom, and Matthew O'Kelly. An active inference model of car following: Advantages and applications. *arXiv preprint arXiv:2303.15201*, 2023.

[17] Ran Wei, Siliang Zeng, Chenliang Li, Alfredo Garcia, and Anthony D McDonald. Robust inverse reinforcement learning through bayesian theory of mind. In *Accepted at ICML 2023 Workshop on Theory of Mind in Communicating Agents*, 2023.

[18] Karl Friston, Thomas FitzGerald, Francesco Rigoli, Philipp Schwartenbeck, and Giovanni Pezzulo. Active inference: a process theory. *Neural computation*, 29(1):1–49, 2017.

[19] Alexander Tschantz, Beren Millidge, Anil K Seth, and Christopher L Buckley. Reinforcement learning through active inference. *arXiv preprint arXiv:2002.12636*, 2020.

[20] Jakob Hohwy. The self-evidencing brain. *Noûs*, 50(2):259–285, 2016.

[21] Morten Henriksen. Variational free energy and economics optimizing with biases and bounded rationality. *Frontiers in Psychology*, 11:549187, 2020.

[22] Karl Friston. The free-energy principle: a unified brain theory? *Nature reviews neuro-*

science, 11(2):127–138, 2010.

[23] Emanuel Todorov and Michael I Jordan. Optimal feedback control as a theory of motor coordination. *Nature neuroscience*, 5(11):1226–1235, 2002.

[24] Manuel Baltieri and Christopher L Buckley. The modularity of action and perception revisited using control theory and active inference. *arXiv preprint arXiv:1806.02649*, 2018.

[25] H von Helmholtz et al. Treatise on physiological optics. *Rochester, NY: Optical Society of America*, 1925.

[26] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

[27] Nicholas Roy, Geoffrey Gordon, and Sebastian Thrun. Finding approximate pomdp solutions through belief compression. *Journal of artificial intelligence research*, 23:1–40, 2005.

[28] Beren Millidge, Alexander Tschantz, Anil K Seth, and Christopher L Buckley. On the relationship between active inference and control as inference. In *International Workshop on Active Inference*, pages 3–11. Springer, 2020.

[29] Lancelot Da Costa, Noor Sajid, Thomas Parr, Karl Friston, and Ryan Smith. The relationship between dynamic programming and active inference: The discrete, finite-horizon case. *arXiv preprint arXiv:2009.08111*, 2020.

[30] Beren Millidge, Alexander Tschantz, Anil Seth, and Christopher Buckley. Understanding the origin of information-seeking exploration in probabilistic objectives for control. *arXiv preprint arXiv:2103.06859*, 2021.

[31] Beren Millidge, Alexander Tschantz, and Christopher L Buckley. Whence the expected free energy? *Neural Computation*, 33(2):447–482, 2021.

[32] Alexander Tschantz, Anil K Seth, and Christopher L Buckley. Learning action-oriented models through active inference. *PLoS computational biology*, 16(4):e1007805, 2020.

[33] Piotr Litwin and Marcin Miłkowski. Unification by fiat: arrested development of predictive processing. *Cognitive Science*, 44(7):e12867, 2020.

[34] Richard D Smallwood and Edward J Sondik. The optimal control of partially observable

markov processes over a finite horizon. *Operations research*, 21(5):1071–1088, 1973.

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

[36] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3):279–292, 1992.

[37] Guy Shani, Joelle Pineau, and Robert Kaplow. A survey of point-based pomdp solvers. *Autonomous Agents and Multi-Agent Systems*, 27(1):1–51, 2013.

[38] Michael Lederman Littman. *Algorithms for sequential decision-making*. Brown University, 1996.

[39] Milos Hauskrecht. Value-function approximations for partially observable markov decision processes. *Journal of artificial intelligence research*, 13:33–94, 2000.

[40] Joelle Pineau, Geoff Gordon, Sebastian Thrun, et al. Point-based value iteration: An anytime algorithm for pomdps. In *Ijcai*, volume 3, pages 1025–1032, 2003.

[41] Hanna Kurniawati, David Hsu, and Wee Sun Lee. Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces. In *Robotics: Science and systems*, volume 2008. Zurich, Switzerland, 2008.

[42] Michael L Littman, Anthony R Cassandra, and Leslie Pack Kaelbling. Learning policies for partially observable environments: Scaling up. In *Machine Learning Proceedings 1995*, pages 362–370. Elsevier, 1995.

[43] Ronald A Howard. Information value theory. *IEEE Transactions on systems science and cybernetics*, 2(1):22–26, 1966.

[44] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

[45] Tomer D Ullman and Joshua B Tenenbaum. Bayesian models of conceptual development: Learning as building models of the world. 2020.

[46] Nathan Lambert, Brandon Amos, Omry Yadan, and Roberto Calandra. Objective mismatch

in model-based reinforcement learning. *arXiv preprint arXiv:2002.04523*, 2020.

[47] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

[48] Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49(2):209–232, 2002.

[49] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31, 2018.

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

[51] Yinlam Chow, Brandon Cui, MoonKyung Ryu, and Mohammad Ghavamzadeh. Variational model-based policy optimization. *arXiv preprint arXiv:2006.05443*, 2020.

[52] Nirbhay Modhe, Harish Kamath, Dhruv Batra, and Ashwin Kalyan. Model-advantage optimization for model-based reinforcement learning. *arXiv preprint arXiv:2106.14080*, 2021.

[53] Christopher Grimm, André Barreto, Satinder Singh, and David Silver. The value equivalence principle for model-based reinforcement learning. *Advances in Neural Information Processing Systems*, 33:5541–5552, 2020.

[54] Alex X Lee, Anusha Nagabandi, Pieter Abbeel, and Sergey Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *Advances in Neural Information Processing Systems*, 33:741–752, 2020.

[55] Nicholas Rhinehart, Jenny Wang, Glen Berseth, John D Co-Reyes, Danijar Hafner, Chelsea Finn, and Sergey Levine. Intrinsic control of variational beliefs in dynamic partially-observed visual environments. In *ICML 2021 Workshop on Unsupervised Reinforcement Learning*, 2021.

[56] Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and

review. *arXiv preprint arXiv:1805.00909*, 2018.

[57] Marc Toussaint, Stefan Harmeling, and Amos Storkey. Probabilistic inference for solving (po) mdps. Technical report, Technical Report EDI-INF-RR-0934, School of Informatics, University of Edinburgh, 2006.

[58] Alec Solway and Matthew M Botvinick. Goal-directed decision making as probabilistic inference: a computational framework and potential neural correlates. *Psychological review*, 119(1):120, 2012.

[59] Emanuel Todorov. Parallels between sensory and motor information processing. *The cognitive neurosciences,*, pages 613–24, 2009.

[60] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[61] Mohammad Ghavamzadeh, Shie Mannor, Joelle Pineau, Aviv Tamar, et al. Bayesian reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 8(5-6): 359–483, 2015.

[62] Michael O'Gordon Duff. *Optimal Learning: Computational procedures for Bayes-adaptive Markov decision processes*. University of Massachusetts Amherst, 2002.

[63] Stephane Ross, Brahim Chaib-draa, and Joelle Pineau. Bayes-adaptive pomdps. *Advances in neural information processing systems*, 20, 2007.

[64] Luisa Zintgraf, Kyriacos Shiarlis, Maximilian Igl, Sebastian Schulze, Yarin Gal, Katja Hofmann, and Shimon Whiteson. Varibad: A very good method for bayes-adaptive deep rl via meta-learning. *arXiv preprint arXiv:1910.08348*, 2019.

[65] Lancelot Da Costa, Thomas Parr, Noor Sajid, Sebastijan Veselic, Victorita Neacsu, and Karl Friston. Active inference on discrete state-spaces: A synthesis. *Journal of Mathematical Psychology*, 99:102447, 2020.

[66] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

[67] Karl Friston, Francesco Rigoli, Dimitri Ognibene, Christoph Mathys, Thomas Fitzgerald, and Giovanni Pezzulo. Active inference and epistemic value. *Cognitive neuroscience*, 6(4): 187–214, 2015.

[68] Noor Sajid, Philip J. Ball, Thomas Parr, and Karl J. Friston. Active inference: Demystified and compared. *Neural Computation*, pages 1–39, 2021.

[69] Beren Millidge. Deep active inference as variational policy gradients. *Journal of Mathematical Psychology*, 96:102348, 2020.

[70] Sebastian Gottwald and Daniel A Braun. The two kinds of free energy and the bayesian revolution. *PLoS computational biology*, 16(12):e1008420, 2020.

[71] Colin Klein. What do predictive coders want? *Synthese*, 195(6):2541–2557, 2018.

[72] Thomas Parr and Karl J Friston. Generalised free energy and active inference. *Biological cybernetics*, 113(5):495–513, 2019.

[73] Miguel Aguilera, Beren Millidge, Alexander Tschantz, and Christopher L Buckley. How particular is the physics of the free energy principle? *Physics of Life Reviews*, 2021.

[74] Karl Friston, Lancelot Da Costa, Dalton AR Sakthivadivel, Conor Heins, Grigorios A Pavliotis, Maxwell Ramstead, and Thomas Parr. Path integrals, particular kinds, and strange things. *arXiv preprint arXiv:2210.12761*, 2022.

[75] Lancelot Da Costa, Karl Friston, Conor Heins, and Grigorios A Pavliotis. Bayesian mechanics for stationary processes. *Proceedings of the Royal Society A*, 477(2256):20210518, 2021.

[76] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[77] Tim Schneider. *Active inference for robotic manipulation*. Technische Universität Darmstadt, 2021.

[78] Sarah Schwöbel, Stefan Kiebel, and Dimitrije Marković. Active inference, belief propagation, and the bethe approximation. *Neural computation*, 30(9):2530–2567, 2018.

[79] Magnus T Koudahl, Wouter M Kouw, and Bert de Vries. On epistemics in expected free

energy for linear gaussian state space models. *Entropy*, 23(12):1565, 2021.

[80] Pietro Mazzaglia, Tim Verbelen, and Bart Dhoedt. Contrastive active inference. *Advances in Neural Information Processing Systems*, 34:13870–13882, 2021.

[81] Pietro Mazzaglia, Tim Verbelen, Ozan Çatal, and Bart Dhoedt. The free energy principle for perception and action: A deep learning perspective. *Entropy*, 24(2):301, 2022.

[82] Karl Friston, Lancelot Da Costa, Danijar Hafner, Casper Hesp, and Thomas Parr. Sophisticated inference. *Neural Computation*, 33(3):713–763, 2021.

[83] Karl Friston. What is optimal about motor control? *Neuron*, 72(3):488–498, 2011.

[84] Harriet Brown, Rick A Adams, Isabel Parees, Mark Edwards, and Karl Friston. Active inference, sensory attenuation and illusions. *Cognitive processing*, 14(4):411–427, 2013.

[85] Anatol G Feldman. New insights into action–perception coupling. *Experimental brain research*, 194(1):39–58, 2009.

[86] Karl Friston, Spyridon Samothrakis, and Read Montague. Active inference and agency: optimal control without cost functions. *Biological cybernetics*, 106(8):523–541, 2012.

[87] Laurence Aitchison and Máté Lengyel. With or without you: predictive coding and bayesian inference in the brain. *Current opinion in neurobiology*, 46:219–227, 2017.

[88] Joseph Marino. Predictive coding, variational autoencoders, and biological connections. *Neural Computation*, 34(1):1–44, 2022.

[89] Karl J Friston, Jean Daunizeau, James Kilner, and Stefan J Kiebel. Action and behavior: a free-energy formulation. *Biological cybernetics*, 102(3):227–260, 2010.

[90] Jelle Bruineberg, Julian Kiverstein, and Erik Rietveld. The anticipating brain is not a scientist: the free-energy principle from an ecological-enactive perspective. *Synthese*, 195(6): 2417–2444, 2018.

[91] Wanja Wiese. Action is enabled by systematic misrepresentations. *Erkenntnis*, 82(6):1233–1252, 2017.

[92] Andy Clark. Beyond desire? agency, choice, and the predictive mind. *Australasian Journal of Philosophy*, 98(1):1–15, 2020.

[93] Maxwell JD Ramstead, Michael D Kirchhoff, and Karl J Friston. A tale of two densities: Active inference is enactive inference. *Adaptive Behavior*, 28(4):225–239, 2020.

[94] Martin J Pickering and Andy Clark. Getting ahead: forward models and their place in cognitive architecture. *Trends in cognitive sciences*, 18(9):451–456, 2014.

[95] Joshua M Martin, Mark Solms, and Philipp Sterzer. Useful misrepresentation: Perception as embodied proactive inference. *Trends in Neurosciences*, 44(8):619–628, 2021.

[96] Colin Klein. *A humean challenge to predictive coding*. Bloomsbury Press, 2020.

[97] Michael D Kirchhoff and Thomas van Es. A universal ethology challenge to the free energy principle: species of inference and good regulators. *Biology & Philosophy*, 36(2):1–24, 2021.

[98] Matteo Colombo and Cory Wright. Explanatory pluralism: An unrewarding prediction error for free energy theorists. *Brain and Cognition*, 112:3–12, 2017.

[99] Manuel Baltieri and Christopher L Buckley. Nonmodular architectures of cognitive systems based on active inference. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[100] Danijar Hafner, Pedro A Ortega, Jimmy Ba, Thomas Parr, Karl Friston, and Nicolas Heess. Action and perception as divergence minimization. *arXiv preprint arXiv:2009.01791*, 2020.

[101] Samuel J Gershman, Eric J Horvitz, and Joshua B Tenenbaum. Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. *Science*, 349(6245): 273–278, 2015.

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

[103] Krishnamurthy Dvijotham and Emanuel Todorov. Inverse optimal control with linearly-solvable mdps. In *ICML*, 2010.

[104] John Rust. Optimal replacement of gmc bus engines: An empirical model of harold zurcher. *Econometrica: Journal of the Econometric Society*, pages 999–1033, 1987.

[105] Chris Baker, Rebecca Saxe, and Joshua Tenenbaum. Bayesian theory of mind: Modeling

joint belief-desire attribution. In *Proceedings of the annual meeting of the cognitive science society*, volume 33, 2011.

[106] Judea Pearl. Causal inference in statistics: An overview. *Statistics surveys*, 3:96–146, 2009.

[107] Pedro A Ortega and Daniel A Braun. A minimum relative entropy principle for learning and acting. *Journal of Artificial Intelligence Research*, 38:475–511, 2010.

[108] Zhengwei Wu, Paul Schrater, and Xaq Pitkow. Inverse rational control: Inferring what you think from how you forage. *arXiv preprint arXiv:1805.09864*, 2018.

[109] Siddharth Reddy, Anca D Dragan, and Sergey Levine. Where do you think you're going?: Inferring beliefs about dynamics from behavior. *arXiv preprint arXiv:1805.08010*, 2018.

[110] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, Jan Peters, et al. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1-2):1–179, 2018.

[111] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

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

[113] Deepak Ramachandran and Eyal Amir. Bayesian inverse reinforcement learning. In *IJCAI*, volume 7, pages 2586–2591, 2007.

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

[115] Brian D Ziebart. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Carnegie Mellon University, 2010.

[116] Kuno Kim, Shivam Garg, Kirankumar Shiragur, and Stefano Ermon. Reward identification in inverse reinforcement learning. In *International Conference on Machine Learning*, pages 5496–5505. PMLR, 2021.

[117] Siliang Zeng, Chenliang Li, Alfredo Garcia, and Mingyi Hong. Maximum-likelihood in-

verse reinforcement learning with finite-time guarantees. In *Decision Awareness in Reinforcement Learning Workshop at ICML 2022*.

[118] Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248*, 2017.

[119] Tung Phan-Minh, Forbes Howington, Ting-Sheng Chu, Sang Uk Lee, Momchil S Tomov, Nanxiang Li, Caglayan Dicle, Samuel Findler, Francisco Suarez-Ruiz, Robert Beaudoin, et al. Driving in real life with inverse reinforcement learning. *arXiv preprint arXiv:2206.03004*, 2022.

[120] JD Choi and Kee-Eung Kim. Inverse reinforcement learning in partially observable environments. *Journal of Machine Learning Research*, 12:691–730, 2011.

[121] Rafael Rafailov, Tianhe Yu, Aravind Rajeswaran, and Chelsea Finn. Visual adversarial imitation learning using variational models. *Advances in Neural Information Processing Systems*, 34:3016–3028, 2021.

[122] Neha Das, Sarah Bechtle, Todor Davchev, Dinesh Jayaraman, Akshara Rai, and Franziska Meier. Model-based inverse reinforcement learning from visual demonstrations. *arXiv preprint arXiv:2010.09034*, 2020.

[123] Pierre-Luc Bacon, Florian Schäfer, Clement Gehring, Animashree Anandkumar, and Emma Brunskill. A lagrangian method for inverse problems in reinforcement learning. In *Optimization in RL workshop at NeurIPS*, volume 2019, 2019.

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

[125] Matthew Golub, Steven Chase, and Byron Yu. Learning an internal dynamics model from control demonstration. In *International Conference on Machine Learning*, pages 606–614. PMLR, 2013.

[126] Ze Gong and Yu Zhang. 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, volume 34, pages 2485–2492, 2020.

[127] Masahiro Kohjima, Tatsushi Matsubayashi, and Hiroshi Sawada. Generalized inverse reinforcement learning with linearly solvable mdp. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 373–388. Springer, 2017.

[128] Minhae Kwon, Saurabh Daptardar, Paul Schrater, and Xaq Pitkow. Inverse rational control with partially observable continuous nonlinear dynamics. *arXiv preprint arXiv:2009.12576*, 2020.

[129] Felix Schmitt, Hans-Joachim Bieg, Michael Herman, and Constantin A Rothkopf. I see what you see: Inferring sensor and policy models of human real-world motor behavior. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[130] Tanmay Gangwani, Joel Lehman, Qiang Liu, and Jian Peng. Learning belief representations for imitation learning in pomdps. In *Uncertainty in Artificial Intelligence*, pages 1061–1071. PMLR, 2020.

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

[132] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

[133] Anna N Rafferty, Michelle M LaMar, and Thomas L Griffiths. Inferring learners' knowledge from their actions. *Cognitive Science*, 39(3):584–618, 2015.

[134] Zhengwei Wu, Minhae Kwon, Saurabh Daptardar, Paul Schrater, and Xaq Pitkow. Rational thoughts in neural codes. *Proceedings of the National Academy of Sciences*, 117(47):29311–29320, 2020.

[135] Ryan Smith, Paul Badcock, and Karl J Friston. Recent advances in the application of predictive coding and active inference models within clinical neuroscience. *Psychiatry and Clinical Neurosciences*, 75(1):3–13, 2021.

[136] Takaki Makino and Johane Takeuchi. Apprenticeship learning for model parameters of

partially observable environments. *arXiv preprint arXiv:1206.6484*, 2012.

[137] Minghuan Liu, Tairan He, Minkai Xu, and Weinan Zhang. Energy-based imitation learning. *arXiv preprint arXiv:2004.09395*, 2020.

[138] Daniel Jarrett, Ioana Bica, and Mihaela van der Schaar. Strictly batch imitation learning by energy-based distribution matching. *Advances in Neural Information Processing Systems*, 33:7354–7365, 2020.

[139] Pete Florence, Corey Lynch, Andy Zeng, Oscar A Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Conference on Robot Learning*, pages 158–168. PMLR, 2022.

[140] Algebraic Pavel (https://math.stackexchange.com/users/90996/algebraic pavel). Solving $ax = b$ when x and b are given. Mathematics Stack Exchange. URL `https://math.stackexchange.com/q/1171850`. URL:https://math.stackexchange.com/q/1171850 (version: 2015-03-02).

[141] Guillaume Bouchard and Bill Triggs. The tradeoff between generative and discriminative classifiers. In *16th IASC International Symposium on Computational Statistics (COMPSTAT'04)*, pages 721–728, 2004.

[142] Bradley Efron. Bayes, oracle bayes and empirical bayes. *Statistical science*, 34(2):177–201, 2019.

[143] Matthias Seeger. Input-dependent regularization of conditional density models. Technical report, 2000.

[144] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. *Advances in neural information processing systems*, 17, 2004.

[145] Aurick Zhou and Sergey Levine. Bayesian adaptation for covariate shift. *Advances in Neural Information Processing Systems*, 34:914–927, 2021.

[146] Christopher M. Bishop and Julia Lasserre. Generative or discriminative? getting the best of both worlds. *Bayesian statistics*, 8(3):3–24, 2007.

[147] Andrew Ng and Michael Jordan. On discriminative vs. generative classifiers: A comparison

of logistic regression and naive bayes. *Advances in neural information processing systems*, 14, 2001.

[148] Terence J O'neill. Normal discrimination with unclassified observations. *Journal of the American Statistical Association*, 73(364):821–826, 1978.

[149] World Health Organization. Global status report on road safety. Technical report, World Health Organization, 2015.

[150] Myra Blanco, Jon Atwood, Sheldon M Russell, Tammy Trimble, Julie A McClafferty, and Miguel A Perez. Automated vehicle crash rate comparison using naturalistic data. Technical report, Virginia Tech Transportation Institute, 2016.

[151] Eric R Teoh and David G Kidd. Rage against the machine? google's self-driving cars versus human drivers. *Journal of safety research*, 63:57–60, 2017.

[152] Victoria A Banks, Katherine L Plant, and Neville A Stanton. Driver error or designer error: Using the perceptual cycle model to explore the circumstances surrounding the fatal tesla crash on 7th may 2016. *Safety science*, 108:278–285, 2018.

[153] Trent W Victor, Emma Tivesten, Pär Gustavsson, Joel Johansson, Fredrik Sangberg, and Mikael Ljung Aust. Automation expectation mismatch: incorrect prediction despite eyes on threat and hands on wheel. *Human factors*, 60(8):1095–1116, 2018.

[154] Anthony D McDonald, Hananeh Alambeigi, Johan Engström, Gustav Markkula, Tobias Vogelpohl, Jarrett Dunne, and Norbert Yuma. Toward computational simulations of behavior during automated driving takeovers: a review of the empirical and modeling literatures. *Human factors*, 61(4):642–688, 2019.

[155] John D Lee, Christopher D Wickens, Yili Liu, and Linda Ng Boyle. *Designing for people: An introduction to human factors engineering*. CreateSpace, 2017.

[156] Gustav Markkula. *Driver behavior models for evaluating automotive active safety: From neural dynamics to vehicle dynamics*. Chalmers University of Technology, 2015.

[157] Christian Roesener, Johannes Hiller, Hendrik Weber, and Lutz Eckstein. How safe is automated driving? human driver models for safety performance assessment. In *2017 IEEE 20th*

International Conference on Intelligent Transportation Systems (ITSC), pages 1–7. IEEE, 2017.

[158] Jonas Bärgman, Christian-Nils Boda, and Marco Dozza. Counterfactual simulations applied to shrp2 crashes: The effect of driver behavior models on safety benefit estimations of intelligent safety systems. *Accident Analysis & Prevention*, 102:165–180, 2017.

[159] Gustav Markkula, Richard Romano, Ruth Madigan, Charles W Fox, Oscar T Giles, and Natasha Merat. Models of human decision-making as tools for estimating and optimizing impacts of vehicle automation. *Transportation research record*, 2672(37):153–163, 2018.

[160] David N Lee. A theory of visual control of braking based on information about time-to-collision. *Perception*, 5(4):437–459, 1976.

[161] Mohammad Saifuzzaman and Zuduo Zheng. Incorporating human-factors in car-following models: a review of recent developments and research needs. *Transportation research part C: emerging technologies*, 48:379–403, 2014.

[162] Gustav Markkula, Johan Engström, Johan Lodin, Jonas Bärgman, and Trent Victor. A farewell to brake reaction times? kinematics-dependent brake response in naturalistic rear-end emergencies. *Accident Analysis & Prevention*, 95:209–226, 2016.

[163] Malin Svärd, Gustav Markkula, Johan Engström, Fredrik Granum, and Jonas Bärgman. A quantitative driver model of pre-crash brake onset and control. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 61, pages 339–343. SAGE Publications Sage CA: Los Angeles, CA, 2017.

[164] Johan Engström, Gustav Markkula, Qingwan Xue, and Natasha Merat. Simulating the effect of cognitive load on braking responses in lead vehicle braking scenarios. *IET Intelligent Transport Systems*, 12(6):427–433, 2018.

[165] Qingwan Xue, Gustav Markkula, Xuedong Yan, and Natasha Merat. Using perceptual cues for brake response to a lead vehicle: Comparing threshold and accumulator models of visual looming. *Accident Analysis & Prevention*, 118:114–124, 2018.

[166] Malin Svärd, Gustav Markkula, Jonas Bärgman, and Trent Victor. Computational modeling

of driver pre-crash brake response, with and without off-road glances: Parameterization using real-world crashes and near-crashes. *Accident Analysis & Prevention*, 163:106433, 2021.

[167] Giulio Bianchi Piccinini, Esko Lehtonen, Fabio Forcolin, Johan Engström, Deike Albers, Gustav Markkula, Johan Lodin, and Jesper Sandin. How do drivers respond to silent automation failures? driving simulator study and comparison of computational driver braking models. *Human factors*, 62(7):1212–1229, 2020.

[168] Jami Pekkanen, Otto Lappi, Paavo Rinkkala, Samuel Tuhkanen, Roosa Frantsi, and Heikki Summala. A computational model for driver's cognitive state, visual perception and intermittent attention in a distracted car following task. *Royal Society open science*, 5(9):180194, 2018.

[169] Peter A Hancock, Tara Kajaks, Jeff K Caird, Mark H Chignell, Sachi Mizobuchi, Peter C Burns, Jing Feng, Geoff R Fernie, Martin Lavallière, Ian Y Noy, et al. Challenges to human drivers in increasingly automated vehicles. *Human factors*, 62(2):310–328, 2020.

[170] Pär Gustavsson, Trent W Victor, Joel Johansson, Emma Tivesten, Regina Johansson, and L Aust. What were they thinking? subjective experiences associated with automation expectation mismatch. In *Proceedings of the 6th Driver Distraction and Inattention conference, Gothenburg, Sweden*, pages 15–17, 2018.

[171] Bobbie D Seppelt and Trent W Victor. Potential solutions to human factors challenges in road vehicle automation. In *Road vehicle automation 3*, pages 131–148. Springer, 2016.

[172] Alberto Morando, Trent Victor, Klaus Bengler, and Marco Dozza. Users' response to critical situations in automated driving: Rear-ends, sideswipes, and false warnings. *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[173] Natasha Merat, Bobbie Seppelt, Tyron Louw, Johan Engström, John D Lee, Emma Johansson, Charles A Green, Satoshi Katazaki, Chris Monk, Makoto Itoh, et al. The "out-of-the-loop" concept in automated driving: Proposed definition, measures and implications. *Cognition, Technology & Work*, 21(1):87–98, 2019.

[174] Gustav Markkula, Erwin Boer, Richard Romano, and Natasha Merat. Sustained sensorimotor control as intermittent decisions about prediction errors: Computational framework and application to ground vehicle steering. *Biological cybernetics*, 112(3):181–207, 2018.

[175] Johan Engström, Jonas Bärgman, Daniel Nilsson, Bobbie Seppelt, Gustav Markkula, Giulio Bianchi Piccinini, and Trent Victor. Great expectations: a predictive processing account of automobile driving. *Theoretical issues in ergonomics science*, 19(2):156–194, 2018.

[176] Philipp Schwartenbeck, Thomas HB FitzGerald, Christoph Mathys, Ray Dolan, and Karl Friston. The dopaminergic midbrain encodes the expected certainty about desired outcomes. *Cerebral cortex*, 25(10):3434–3445, 2015.

[177] Ryan Smith, Rayus Kuplicki, Justin Feinstein, Katherine L Forthman, Jennifer L Stewart, Martin P Paulus, Tulsa 1000 investigators, and Sahib S Khalsa. A bayesian computational model reveals a failure to adapt interoceptive precision estimates across depression, anxiety, eating, and substance use disorders. *PLOS Computational Biology*, 16(12):e1008484, 2020.

[178] Ray Fuller. Towards a general theory of driver behaviour. *Accident analysis & prevention*, 37(3):461–472, 2005.

[179] Harriet Brown, Karl J Friston, and Sven Bestmann. Active inference, attention, and motor preparation. *Frontiers in psychology*, 2:218, 2011.

[180] Sebastian Bitzer, Hame Park, Felix Blankenburg, and Stefan J Kiebel. Perceptual decision making: drift-diffusion model is equivalent to a bayesian model. *Frontiers in human neuroscience*, 8:102, 2014.

[181] Thomas Parr and Karl J Friston. Uncertainty, epistemics and active inference. *Journal of The Royal Society Interface*, 14(136):20170376, 2017.

[182] Hananeh Alambeigi and Anthony D McDonald. A bayesian regression analysis of the effects of alert presence and scenario criticality on automated vehicle takeover performance. *Human factors*, page 00187208211010004, 2021.

[183] Anthony D McDonald, Abhijit Sarkar, Jeffrey Hickman, Hananeh Alambeigi, Tobias Vo-

gelpohl, and Gustav Markkula. Modeling driver behavior during automated vehicle platooning failures. 2021.

[184] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[185] Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. Finding a" kneedle" in a haystack: Detecting knee points in system behavior. In *2011 31st international conference on distributed computing systems workshops*, pages 166–171. IEEE, 2011.

[186] John D Lee and Katrina A See. Trust in automation: Designing for appropriate reliance. *Human factors*, 46(1):50–80, 2004.

[187] Mica R Endsley. Toward a theory of situation awareness in dynamic systems. *Human factors*, 37(1):32–64, 1995.

[188] Gustav Markkula, Ola Benderius, and Mattias Wahde. Comparing and validating models of driver steering behaviour in collision avoidance and vehicle stabilisation. *Vehicle system dynamics*, 52(12):1658–1680, 2014.

[189] Alexander Eriksson, VA Banks, and NA Stanton. Transition to manual: Comparing simulator with on-road control transitions. *Accident Analysis & Prevention*, 102:227–234, 2017.

[190] Oliver Scheel, Luca Bergamini, Maciej Wolczyk, Błażej Osiński, and Peter Ondruska. Urban driver: Learning to drive from real-world demonstrations using policy gradients. In *Conference on Robot Learning*, pages 718–728. PMLR, 2022.

[191] John M Scanlon, Kristofer D Kusano, Tom Daniel, Christopher Alderson, Alexander Ogle, and Trent Victor. Waymo simulated driving behavior in reconstructed fatal crashes within an autonomous vehicle operating domain. *Accident Analysis & Prevention*, 163:106454, 2021.

[192] Jonas Bärgman, Christian-Nils Boda, and Marco Dozza. Counterfactual simulations applied to SHRP2 crashes: The effect of driver behavior models on safety benefit estimations of intelligent safety systems. *Accident Analysis & Prevention*, 102:165–180, 2017. ISSN 0001-4575. doi: https://doi.org/10.1016/j.aap.2017.03.003. URL `https://www.sciencedirect.com/science/article/pii/S000145751730101X`.

[193] Nicholas Rhinehart, Rowan McAllister, Kris Kitani, and Sergey Levine. Precog: Prediction conditioned on goals in visual multi-agent settings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2821–2830, 2019.

[194] Gustav Markkula, Yi-Shin Lin, Aravinda Ramakrishnan Srinivasan, Jac Billington, Matteo Leonetti, Amir Hossein Kalantari, Yue Yang, Yee Mun Lee, Ruth Madigan, and Natasha Merat. Explaining human interactions on the road requires large-scale integration of psychological theory. 2022.

[195] Tilman Räukur, Anson Ho, Stephen Casper, and Dylan Hadfield-Menell. Toward transparent ai: A survey on interpreting the inner structures of deep neural networks. *arXiv preprint arXiv:2207.13243*, 2022.

[196] Hananeh Alambeigi, Anthony D. McDonald, and Srinivas R. Tankasala. Crash themes in automated vehicles: A topic modeling analysis of the california department of motor vehicles automated vehicle crash database, 2020. URL `https://arxiv.org/abs/2001.11087`.

[197] Fjollë Novakazi, Mikael Johansson, Helena Strömberg, and MariAnne Karlsson. Levels of what? investigating drivers' understanding of different levels of automation in vehicles. *Journal of Cognitive Engineering and Decision Making*, 15(2-3):116–132, 2021.

[198] Arne Kesting, Martin Treiber, and Dirk Helbing. Agents for traffic simulation. *Multi-agent systems: Simulation and applications*, 5, 2009.

[199] Samer H Hamdar and Hani S Mahmassani. From existing accident-free car-following models to colliding vehicles: exploration and assessment. *Transportation research record*, 2088 (1):45–56, 2008.

[200] Alireza Talebpour and Hani S Mahmassani. Influence of connected and autonomous vehicles on traffic flow stability and throughput. *Transportation Research Part C: Emerging Technologies*, 71:143–163, 2016.

[201] Mofan Zhou, Xiaobo Qu, and Xiaopeng Li. A recurrent neural network based microscopic car following model to predict traffic oscillation. *Transportation research part C: emerging*

technologies, 84:245–264, 2017.

[202] Zhaobin Mo, Rongye Shi, and Xuan Di. A physics-informed deep learning paradigm for car-following models. *Transportation research part C: emerging technologies*, 130:103240, 2021.

[203] Meixin Zhu, Xuesong Wang, and Yinhai Wang. Human-like autonomous car-following model with deep reinforcement learning. *Transportation research part C: emerging technologies*, 97:348–368, 2018.

[204] Raunak Bhattacharyya, Blake Wulfe, Derek Phillips, Alex Kuefler, Jeremy Morton, Ransalu Senanayake, and Mykel Kochenderfer. Modeling human driving behavior through generative adversarial imitation learning. *arXiv preprint arXiv:2006.06412*, 2020.

[205] Jonathan Spencer, Sanjiban Choudhury, Arun Venkatraman, Brian Ziebart, and J Andrew Bagnell. Feedback in imitation learning: The three regimes of covariate shift. *arXiv preprint arXiv:2102.02872*, 2021.

[206] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

[207] Felipe Codevilla, Eder Santana, Antonio M López, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9329–9338, 2019.

[208] Aviral Kumar, Joey Hong, Anikait Singh, and Sergey Levine. Should i run offline reinforcement learning or behavioral cloning? In *International Conference on Learning Representations*, 2021.

[209] Maximilian Igl, Daewoo Kim, Alex Kuefler, Paul Mougin, Punit Shah, Kyriacos Shiarlis, Dragomir Anguelov, Mark Palatucci, Brandyn White, and Shimon Whiteson. Symphony: Learning realistic and diverse agents for autonomous driving simulation. *arXiv preprint arXiv:2205.03195*, 2022.

[210] Simon Suo, Sebastian Regalado, Sergio Casas, and Raquel Urtasun. Trafficsim: Learning to simulate realistic multi-agent behaviors. In *Proceedings of the IEEE/CVF Conference on*

Computer Vision and Pattern Recognition, pages 10400–10409, 2021.

[211] Kenji Doya, Shin Ishii, Alexandre Pouget, and Rajesh PN Rao. *Bayesian brain: Probabilistic approaches to neural coding*. MIT press, 2007.

[212] Johan Engström, Shu-Yuan Liu, Azadeh Dinparastdjadid, and Camelia Simoiu. Modeling road user response timing in naturalistic settings: a surprise-based framework. *arXiv preprint arXiv:2208.08651*, 2022.

[213] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2):1805, 2000.

[214] Vasileia Papathanasopoulou and Constantinos Antoniou. Towards data-driven car-following models. *Transportation Research Part C: Emerging Technologies*, 55:496–509, 2015.

[215] Martin Treiber and Arne Kesting. Microscopic calibration and validation of car-following models–a systematic approach. *Procedia-Social and Behavioral Sciences*, 80:922–939, 2013.

[216] Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 661–668. JMLR Workshop and Conference Proceedings, 2010.

[217] Pim De Haan, Dinesh Jayaraman, and Sergey Levine. Causal confusion in imitation learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[218] Randall D Beer. The dynamics of active categorical perception in an evolved model agent. *Adaptive behavior*, 11(4):209–243, 2003.

[219] Philipp Schwartenbeck, Thomas HB FitzGerald, Christoph Mathys, Ray Dolan, Friedrich Wurst, Martin Kronbichler, and Karl Friston. Optimal inference with suboptimal models: addiction and active bayesian inference. *Medical hypotheses*, 84(2):109–117, 2015.

[220] Wei Zhan, Liting Sun, Di Wang, Haojie Shi, Aubrey Clausse, Maximilian Naumann, Julius Kummerle, Hendrik Konigshof, Christoph Stiller, Arnaud de La Fortelle, et al. Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps. *arXiv preprint arXiv:1910.03088*, 2019.

[221] Fabian Poggenhans, Jan-Hendrik Pauls, Johannes Janosovits, Stefan Orf, Maximilian Naumann, Florian Kuhnt, and Matthias Mayr. Lanelet2: A high-definition map framework for the future of automated driving. In *2018 21st international conference on intelligent transportation systems (ITSC)*, pages 1672–1679. IEEE, 2018.

[222] Moritz Werling, Julius Ziegler, Sören Kammel, and Sebastian Thrun. Optimal trajectory generation for dynamic street scenarios in a frenet frame. In *2010 IEEE International Conference on Robotics and Automation*, pages 987–993. IEEE, 2010.

[223] Soyeon Jung, Ransalu Senanayake, and Mykel J Kochenderfer. A gray box model for characterizing driver behavior. In *SafeAI@ AAAI*, 2022.

[224] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[225] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

[226] George Papamakarios, Eric T Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *J. Mach. Learn. Res.*, 22(57):1–64, 2021.

[227] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, 29, 2016.

[228] Peter Karkus, David Hsu, and Wee Sun Lee. Qmdp-net: Deep learning for planning under partial observability. *Advances in neural information processing systems*, 30, 2017.

[229] Ran Wei. Active inference interaction modeling. `https://github.com/rw422scarlet/interactive_inference`, 2022.

[230] Stuart Armstrong and Sören Mindermann. Impossibility of deducing preferences and rationality from human policy. *arXiv preprint arXiv:1712.05812*, 2017.

[231] Wei Ran, Anthony McDonald, and Alfredo Garcia. World model learning from demonstra-

tions. In *Proceedings of 3rd International Workshop on Active Inference*, 2022.

[232] Dorsa Sadigh, Shankar Sastry, Sanjit A Seshia, and Anca D Dragan. Planning for autonomous cars that leverage effects on human actions. In *Robotics: Science and systems*, volume 2, pages 1–9. Ann Arbor, MI, USA, 2016.

[233] Cédric Colas, Olivier Sigaud, and Pierre-Yves Oudeyer. A hitchhiker's guide to statistical comparisons of reinforcement learning algorithms. *arXiv preprint arXiv:1904.06979*, 2019.

[234] Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in neural information processing systems*, 34:29304–29320, 2021.

[235] Sedan vehicle dimensions. `https://www.mathworks.com/help/driving/ref/sedan.html`. Accessed: 2022-12-15.

[236] Heikki Summala. Hierarchical model of behavioural adaptation and traffic accidents. *Traffic and transport psychology. Theory and application*, 1997.

[237] PA Hancock. Is car following the real question–are equations the answer? *Transportation research part F: traffic psychology and behaviour*, 2(4):197–199, 1999.

[238] Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein. A tutorial on the cross-entropy method. *Annals of operations research*, 134(1):19–67, 2005.

[239] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

[240] Aviv Tamar, Yi Wu, Garrett Thomas, Sergey Levine, and Pieter Abbeel. Value iteration networks. *Advances in neural information processing systems*, 29, 2016.

[241] Trent Victor, Marco Dozza, Jonas Bärgman, Christian-Nils Boda, Johan Engström, Carol Flannagan, John D Lee, and Gustav Markkula. Analysis of naturalistic driving study data: Safer glances, driver inattention, and crash risk. Technical report, 2015.

[242] Shoichiro Yamaguchi, Honda Naoki, Muneki Ikeda, Yuki Tsukada, Shunji Nakano, Ikue Mori, and Shin Ishii. Identification of animal behavioral strategies by inverse reinforcement

learning. *PLoS computational biology*, 14(5):e1006122, 2018.

[243] Stuart Russell. *Human compatible: Artificial intelligence and the problem of control*. Penguin, 2019.

[244] Alex Kuefler, Jeremy Morton, Tim Wheeler, and Mykel Kochenderfer. Imitating driver behavior with generative adversarial networks. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 204–211. IEEE, 2017.

[245] Alex J Chan and Mihaela van der Schaar. Scalable bayesian inverse reinforcement learning. *arXiv preprint arXiv:2102.06483*, 2021.

[246] Divyansh Garg, Shuvam Chakraborty, Chris Cundy, Jiaming Song, and Stefano Ermon. Iq-learn: Inverse soft-q learning for imitation. *Advances in Neural Information Processing Systems*, 34:4028–4039, 2021.

[247] Ilya Kostrikov, Ofir Nachum, and Jonathan Tompson. Imitation learning via off-policy distribution matching. *arXiv preprint arXiv:1912.05032*, 2019.

[248] Sheng Yue, Guanbo Wang, Wei Shao, Zhaofeng Zhang, Sen Lin, Ju Ren, and Junshan Zhang. Clare: Conservative model-based reward learning for offline inverse reinforcement learning. *arXiv preprint arXiv:2302.04782*, 2023.

[249] Siliang Zeng, Chenliang Li, Alfredo Garcia, and Mingyi Hong. Understanding expertise through demonstrations: A maximum likelihood framework for offline inverse reinforcement learning. *arXiv preprint arXiv:2302.07457*, 2023.

[250] Jonathan Chang, Masatoshi Uehara, Dhruv Sreenivas, Rahul Kidambi, and Wen Sun. Mitigating covariate shift in imitation learning via offline data with partial coverage. *Advances in Neural Information Processing Systems*, 34:965–979, 2021.

[251] Daniel Jarrett, Alihan Hüyük, and Mihaela Van Der Schaar. Inverse decision modeling: Learning interpretable representations of behavior. In *International Conference on Machine Learning*, pages 4755–4771. PMLR, 2021.

[252] Gergely Neu, Anders Jonsson, and Vicenç Gómez. A unified view of entropy-regularized markov decision processes. *arXiv preprint arXiv:1705.07798*, 2017.

[253] Adam Gleave and Sam Toyer. A primer on maximum causal entropy inverse reinforcement learning. *arXiv preprint arXiv:2203.11409*, 2022.

[254] Siliang Zeng, Mingyi Hong, and Alfredo Garcia. Structural estimation of markov decision processes in high-dimensional state space with finite-time guarantees. *arXiv preprint arXiv:2210.01282*, 2022.

[255] Seyed Kamyar Seyed Ghasemipour, Richard Zemel, and Shixiang Gu. A divergence minimization perspective on imitation learning methods. In *Conference on Robot Learning*, pages 1259–1277. PMLR, 2020.

[256] Liyiming Ke, Sanjiban Choudhury, Matt Barnes, Wen Sun, Gilwoo Lee, and Siddhartha Srinivasa. Imitation learning as f-divergence minimization. In *International Workshop on the Algorithmic Foundations of Robotics*, pages 313–329. Springer, 2021.

[257] Chelsea Finn, Paul Christiano, Pieter Abbeel, and Sergey Levine. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. *arXiv preprint arXiv:1611.03852*, 2016.

[258] Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *International conference on machine learning*, pages 49–58. PMLR, 2016.

[259] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

[260] Taher Jafferjee, Ehsan Imani, Erin Talvitie, Martha White, and Micheal Bowling. Hallucinating value: A pitfall of dyna-style planning with imperfect environment models. *arXiv preprint arXiv:2006.04363*, 2020.

[261] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33:14129–14142, 2020.

[262] Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel:

Model-based offline reinforcement learning. *Advances in neural information processing systems*, 33:21810–21823, 2020.

[263] Cong Lu, Philip Ball, Jack Parker-Holder, Michael Osborne, and Stephen J Roberts. Revisiting design choices in offline model based reinforcement learning. In *International Conference on Learning Representations*, 2021.

[264] Masatoshi Uehara and Wen Sun. Pessimistic model-based offline reinforcement learning under partial coverage. *arXiv preprint arXiv:2107.06226*, 2021.

[265] Arnab Nilim and Laurent El Ghaoui. Robust control of markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

[266] Garud N Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30 (2):257–280, 2005.

[267] Marc Rigter, Bruno Lacerda, and Nick Hawes. Rambo-rl: Robust adversarial model-based offline reinforcement learning. *arXiv preprint arXiv:2204.12581*, 2022.

[268] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[269] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688, 2011.

[270] Stuart Armstrong and Sören Mindermann. Occam's razor is insufficient to infer the preferences of irrational agents. *Advances in neural information processing systems*, 31, 2018.

[271] Vivek S Borkar. Stochastic approximation with two time scales. *Systems & Control Letters*, 29(5):291–294, 1997.

[272] Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A two-timescale framework for bilevel optimization: Complexity analysis and application to actor-critic. *arXiv preprint arXiv:2007.05170*, 2020.

[273] Anirudh Vemula, Yuda Song, Aarti Singh, J Andrew Bagnell, and Sanjiban Choudhury. The virtues of laziness in model-based rl: A unified objective and algorithms. *arXiv preprint*

arXiv:2303.00694, 2023.

[274] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[275] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

[276] Yilun Du, Shuang Li, Joshua Tenenbaum, and Igor Mordatch. Improved contrastive divergence training of energy based models. *arXiv preprint arXiv:2012.01316*, 2020.

[277] Kavosh Asadi, Dipendra Misra, and Michael Littman. Lipschitz continuity in model-based reinforcement learning. In *International Conference on Machine Learning*, pages 264–273. PMLR, 2018.

[278] Jaedeug Choi and Kee-Eung Kim. Map inference for bayesian inverse reinforcement learning. *Advances in neural information processing systems*, 24, 2011.

[279] Amir-massoud Farahmand, Andre Barreto, and Daniel Nikovski. Value-aware loss function for model-based reinforcement learning. In *Artificial Intelligence and Statistics*, pages 1486–1494. PMLR, 2017.

[280] Christelle Langley, Bogdan Ionut Cirstea, Fabio Cuzzolin, and Barbara J Sahakian. Theory of mind and preference learning at the interface of cognitive science, neuroscience, and ai: A review. *Frontiers in Artificial Intelligence*, page 62, 2022.

[281] Mark K Ho, Michael Littman, James MacGlashan, Fiery Cushman, and Joseph L Austerweil. Showing versus doing: Teaching by demonstration. *Advances in neural information processing systems*, 29, 2016.

[282] Jakob Foerster, Francis Song, Edward Hughes, Neil Burch, Iain Dunning, Shimon Whiteson, Matthew Botvinick, and Michael Bowling. Bayesian action decoder for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 1942–1951. PMLR, 2019.

[283] Reuth Mirsky, Ignacio Carlucho, Arrasy Rahman, Elliot Fosong, William Macke, Mohan

Sridharan, Peter Stone, and Stefano V Albrecht. A survey of ad hoc teamwork research. In *Multi-Agent Systems: 19th European Conference, EUMAS 2022, Düsseldorf, Germany, September 14–16, 2022, Proceedings*, pages 275–293. Springer, 2022.

[284] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

[285] Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? *arXiv preprint arXiv:2211.15657*, 2022.

[286] Juergen Schmidhuber. Reinforcement learning upside down: Don't predict rewards–just map them to actions. *arXiv preprint arXiv:1912.02875*, 2019.

[287] Kyriacos Shiarlis, Joao Messias, and Shimon Whiteson. Rapidly exploring learning trees. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 1541–1548. IEEE, 2017.

[288] Nicolas Zucchet and João Sacramento. Beyond backpropagation: bilevel optimization through implicit differentiation and equilibrium propagation. *Neural Computation*, 34(12): 2309–2346, 2022.

[289] Bilal Piot, Matthieu Geist, and Olivier Pietquin. Boosted and reward-regularized classification for apprenticeship learning. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 1249–1256, 2014.

[290] Lawrence Chan, Andrew Critch, and Anca Dragan. Human irrationality: both bad and good for reward inference. *arXiv preprint arXiv:2111.06956*, 2021.

[291] Joey Hong, Kush Bhatia, and Anca Dragan. On the sensitivity of reward inference to misspecified human models. *arXiv preprint arXiv:2212.04717*, 2022.

[292] Neil Rabinowitz, Frank Perbet, Francis Song, Chiyuan Zhang, SM Ali Eslami, and Matthew Botvinick. Machine theory of mind. In *International conference on machine learning*, pages

4218–4227. PMLR, 2018.

[293] Dylan Hadfield-Menell, Stuart J Russell, Pieter Abbeel, and Anca Dragan. Cooperative inverse reinforcement learning. *Advances in neural information processing systems*, 29, 2016.

[294] Brandon Amos. Tutorial on amortized optimization for learning to optimize over continuous domains. *arXiv preprint arXiv:2202.00665*, 2022.

[295] Marcel Binz, Ishita Dasgupta, Akshay Jagadish, Matthew Botvinick, Jane X Wang, and Eric Schulz. Meta-learned models of cognition. *arXiv preprint arXiv:2304.06729*, 2023.

[296] Pedro A Ortega, Jane X Wang, Mark Rowland, Tim Genewein, Zeb Kurth-Nelson, Razvan Pascanu, Nicolas Heess, Joel Veness, Alex Pritzel, Pablo Sprechmann, et al. Meta-learning of sequential strategies. *arXiv preprint arXiv:1905.03030*, 2019.

[297] Vladimir Mikulik, Grégoire Delétang, Tom McGrath, Tim Genewein, Miljan Martic, Shane Legg, and Pedro Ortega. Meta-trained agents implement bayes-optimal agents. *Advances in neural information processing systems*, 33:18691–18703, 2020.

[298] Andrew Campbell, Yuyang Shi, Thomas Rainforth, and Arnaud Doucet. Online variational filtering and parameter learning. *Advances in Neural Information Processing Systems*, 34: 18633–18645, 2021.

[299] Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, and Lawrence Carin. Cyclical annealing schedule: A simple approach to mitigating KL vanishing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 240–250, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1021. URL https://aclanthology.org/N19-1021.

[300] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[301] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, An-

dreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. pages 8024–8035, 2019.

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

APPENDIX A

APPENDIX FOR CHAPTERS 2 AND 3

A.1 Active Inference Optimal Perception Derivation (section 2.4.2.1)

We find the optimal variational posterior $Q(s_{1:T}|\pi)$ by taking the gradient of the free energy functional $\mathcal{F}(o_{1:t}, Q|\pi)$ and setting it to zero. The gradient for a specific element of vector $Q(s_\tau|\pi)$ for $\tau \leq t$ is:

$$\begin{aligned}
\nabla_{Q(s_\tau|\pi)}\mathcal{F}(o_{1:\tau}, Q) &= \nabla_{Q(s_\tau|\pi)}\mathbb{E}_{Q(s_\tau|\pi)}[\log Q(s_\tau|\pi)] - \nabla_{Q(s_\tau|\pi)}\mathbb{E}_{Q(s_\tau|\pi)}[\log P(o_\tau|s_\tau)] \\
&\quad - \nabla_{Q(s_\tau|\pi)}\mathbb{E}_{Q(s_{\tau-1}|\pi)Q(s_\tau|\pi)}[\log P(s_\tau|s_{\tau-1}, \pi)] \\
&\quad - \nabla_{Q(s_\tau|\pi)}\mathbb{E}_{Q(s_\tau|\pi)Q(s_{\tau+1}|\pi)}[\log P(s_{\tau+1}|s_\tau, \pi)] \\
&= \log Q(s_\tau|\pi) + 1 - \log P(o_\tau|s_\tau) \\
&\quad - \mathbb{E}_{Q(s_{\tau-1}|\pi)}[\log P(s_\tau|s_{\tau-1}, \pi)] - \mathbb{E}_{Q(s_{\tau+1}|\pi)}[\log P(s_{\tau+1}|s_\tau, \pi)]
\end{aligned} \tag{A.1}$$

For $\tau > t$, we have not made any observations, so the $\log P(o_\tau|s_\tau)$ term does not exist. We represent this using an indicator function.

Setting the gradient to zero when $Q(s_\tau|\pi)$ for all $\tau \in 1, ..., T$ have been optimized, we have:

$$\begin{aligned}
\log Q(s_\tau|\pi) = \mathbb{I}[\tau \leq t] \log P(o_\tau|s_\tau) + \mathbb{E}_{Q^*(s_{\tau-1}|\pi)}[\log P(s_\tau|s_{\tau-1}, \pi)] \\
+ \mathbb{E}_{Q^*(s_{\tau+1}|\pi)}[\log P(s_{\tau+1}|s_\tau, \pi)] - 1
\end{aligned} \tag{A.2}$$

A.2 Belief Equivalence KL Divergence Gradient Derivation (3.19)

$$\begin{aligned}\nabla_{\theta_{11}} D_{KL} = &-\mathbb{E}_{b(s_t)} \underbrace{\left[\nabla_{\theta_{11}} \log P(o_t|s_t;\theta_{11}) \sum_{s_{t-1}} P(s_t|s_{t-1},a_{t-1};\theta_{12}) b(s_{t-1})\right]}_{\text{T1}} \\ &+ \mathbb{E}_{b(s_t)} \underbrace{\left[\nabla_{\theta_{11}} \log \sum_{s_t} P(o_t|s_t;\theta_{11}) \sum_{s_{t-1}} P(s_t|s_{t-1},a_{t-1};\theta_{12}) b(s_{t-1})\right]}_{\text{T2}}\end{aligned} \tag{A.3}$$

$$\begin{aligned}\text{T1} &= \frac{\nabla_{\theta_{11}} P(o_t|s_t;\theta_{11}) \sum_{s_{t-1}} P(s_t|s_{t-1},a_{t-1};\theta_{12}) b(s_{t-1})}{P(o_t|s_t;\theta_{11}) \sum_{s_{t-1}} P(s_t|s_{t-1},a_{t-1};\theta_{12}) b(s_{t-1})} \\ &= \frac{\nabla_{\theta_{11}} P(o_t|s_t;\theta_{11})}{P(o_t|s_t;\theta_{11})} \\ \text{T2} &= \frac{1}{Z} \nabla_{\theta_{11}} \sum_{s_t} P(o_t|s_t;\theta_{11}) \sum_{s_{t-1}} P(s_t|s_{t-1},a_{t-1};\theta_{12}) b(s_{t-1}) \\ &= \frac{1}{Z} \sum_{s_t} \sum_{s_{t-1}} P(s_t|s_{t-1},a_{t-1};\theta_{12}) b(s_{t-1}) \nabla_{\theta_{11}} P(o_t|s_t;\theta_{11})\end{aligned} \tag{A.4}$$

Plugging back T1 and T2, we have:

$$\begin{aligned}\nabla_{\theta_{11}} D_{KL} = &-\mathbb{E}_{b(s_t)} \left[\frac{\nabla_{\theta_{11}} P(o_t|s_t;\theta_{11})}{P(o_t|s_t;\theta_{11})}\right] \\ &+ \frac{1}{Z} \sum_{s_t} \sum_{s_{t-1}} P(s_t|s_{t-1},a_{t-1};\theta_{12}) b(s_{t-1}) \nabla_{\theta_{11}} P(o_t|s_t;\theta_{11})\end{aligned} \tag{A.5}$$

Similarly, we define T1 and T2 for the KL divergence gradient w.r.t. the transition parameters:

$$\begin{aligned}\nabla_{\theta_{12}} D_{KL} = &-\mathbb{E}_{b(s_t)} \underbrace{\left[\nabla_{\theta_{12}} \log P(o_t|s_t;\theta_{11}) \sum_{s_{t-1}} P(s_t|s_{t-1},a_{t-1};\theta_{12}) b(s_{t-1})\right]}_{\text{T1}} \\ &+ \mathbb{E}_{b(s_t)} \underbrace{\left[\nabla_{\theta_{12}} \log \sum_{s_t} P(o_t|s_t;\theta_{11}) \sum_{s_{t-1}} P(s_t|s_{t-1},a_{t-1};\theta_{12}) b(s_{t-1})\right]}_{\text{T2}}\end{aligned} \tag{A.6}$$

$$
\begin{aligned}
\text{T1} &= \frac{\nabla_{\theta_{12}} P(o_t|s_t;\theta_{11}) \sum_{s_{t-1}} P(s_t|s_{t-1},a_{t-1};\theta_{12}) b(s_{t-1})}{P(o_t|s_t;\theta_{11}) \sum_{s_{t-1}} P(s_t|s_{t-1},a_{t-1};\theta_{12}) b(s_{t-1})} \\
&= \frac{\sum_{s_{t-1}} b(s_{t-1}) \nabla_{\theta_{12}} P(s_t|s_{t-1},a_{t-1};\theta_{12})}{\sum_{s_{t-1}} P(s_t|s_{t-1},a_{t-1};\theta_{12}) b(s_{t-1})} \\
\text{T2} &= \frac{1}{Z} \nabla_{\theta_{12}} \sum_{s_t} P(o_t|s_t;\theta_{11}) \sum_{s_{t-1}} P(s_t|s_{t-1},a_{t-1};\theta_{12}) b(s_{t-1}) \\
&= \frac{1}{Z} \sum_{s_t} \sum_{s_{t-1}} P(o_t|s_t;\theta_{11}) b(s_{t-1}) \nabla_{\theta_{12}} P(s_t|s_{t-1},a_{t-1};\theta_{12})
\end{aligned}
\tag{A.7}
$$

Plugging back T1 and T2, we have:

$$
\begin{aligned}
\nabla_{\theta_{12}} D_{KL} = &-\mathbb{E}_{b(s_t)} \left[\frac{\sum_{s_{t-1}} b(s_{t-1}) \nabla_{\theta_{12}} P(s_t|s_{t-1},a_{t-1};\theta_{12})}{\sum_{s_{t-1}} P(s_t|s_{t-1},a_{t-1};\theta_{12}) b(s_{t-1})} \right] \\
&+ \frac{1}{Z} \sum_{s_t} \sum_{s_{t-1}} P(o_t|s_t;\theta_{11}) b(s_{t-1}) \nabla_{\theta_{12}} P(s_t|s_{t-1},a_{t-1};\theta_{12})
\end{aligned}
\tag{A.8}
$$

APPENDIX B

APPENDIX FOR CHAPTER 4

B.1 Evidence Lower Bound Derivation

The log-marginal likelihood lower bound, also called the evidence lower bound (ELBO) [66], follows from standard derivation of latent variable models. Introducing a variational distribution $Q(\theta)$, we lower bound the log-marginal likelihood using Jensen's inequality:

$$\begin{aligned}
\mathcal{L}(a_{1:T}|o_{1:T}) &= \log \int P(\theta) \prod_{t=1}^{T} P(a_t|o_{1:t}, \theta) d\theta \\
&= \log \int Q(\theta) \frac{P(\theta)}{Q(\theta)} \prod_{t=1}^{T} P(a_t|o_{1:t}, \theta) d\theta \\
&\geq \mathbb{E}_{Q(\theta)}[\log \frac{P(\theta)}{Q(\theta)} \prod_{t=1}^{T} P(a_t|o_{1:t}, \theta)] \\
&= \mathbb{E}_{Q(\theta)}[\sum_{t=1}^{T} \log P(a_t|o_{1:t}, \theta)] - D_{KL}[Q(\theta)||P(\theta)]
\end{aligned} \tag{B.1}$$

ELBO optimization is difficult due to the presence of local optima. Prior works suggest using KL-annealing to find good initialization [299]. We initialized and prior with a standard normal distribution and used the same randomly initialized posterior parameters for all drives. We optimized the prior and the posterior simultaneously for $10,000$ iterations using the Adam optimizer [300] with learning rate 0.01. We used KL-annealing in the first $5,000$ iterations by adding a coefficient to the negative KL divergence term and increasing it linearly from 0 to 1. Our optimization procedure was implemented in PyTorch [301].

B.2 Precision Update Derivation

The addition of the precision parameter γ leads to the following factorization of the generative and inference models [18]:

$$\begin{aligned} P(o_{1:t}, s_{1:T}, \pi, \gamma) &= P(o_{1:t}|s_{1:t})P(s_{1:T}|\pi)P(\pi|\gamma)P(\gamma) \\ Q(s_{1:T}, \pi, \gamma) &= \prod_{t=1}^{T} Q(s_t|\pi)Q(\pi)Q(\gamma) \end{aligned} \tag{B.2}$$

The action prior depends on γ via:

$$P(\pi|\gamma) \propto \exp\left(-\gamma \mathcal{G}(\pi|Q^*)\right) \tag{B.3}$$

The perception-action loop is augmented with an additional step. Between every time step, the agent first computes the optimal state estimate by minimizing $\mathcal{F}(o_{1:t}, Q|\pi)$. It then updates the precision estimate using $Q(\pi)$ from the last time step with the current state estimates. Finally, it computes the updated policy with the new precision estimate before generating an action.

To obtain the new precision estimate, we differentiate the free energy function and set it to zero. The approximate posterior over γ has the form:

$$\begin{aligned} Q(\gamma) &\propto \exp\left(\log P(\gamma) + \mathbb{E}_{Q(\pi)}\left[\log P(\pi|\gamma)\right]\right) \\ &= \exp\left(\log P(\gamma) - \mathbb{E}_{Q(\pi)}\left[\gamma \mathcal{G}(\pi|Q^*)\right] + \mathbb{E}_{P(\pi|\gamma)}\left[\gamma \mathcal{G}(\pi|Q^*)\right]\right) \end{aligned} \tag{B.4}$$

Modeling the prior distribution over γ as a Gamma distribution $P(\gamma) = \Gamma(\alpha, \beta)$ with a fixed α, the prior expected precision is $\frac{\alpha}{\beta}$. Using the Gamma conjugate posterior property, we can show that the expected precision is:

$$\gamma' = \frac{\alpha}{\beta + \mathbb{E}_{Q(\pi)}\left[\mathcal{G}(\pi|Q^*)\right] - \mathbb{E}_{P(\pi)}\left[\mathcal{G}(\pi|Q^*)\right]} \tag{B.5}$$

where $P(\pi)$ is the EFE prior from the last time step. This result is intuitive as an increase in

expected free energy compared to the prior decreases the precision parameters. This decreases the concentration of the action probability and leads to a reduced commitment to the previously pursued course of actions.

APPENDIX C

APPENDIX FOR CHAPTER 5

C.1 BC Implementation

For BC-MLP, we used a two-layer MLP network with ReLU activation and 40 hidden units in each layer. For BC-RNN, we used a two-layer MLP network on top of a single-layer GRU network with ReLU activation and 30 hidden units in each layer. The GRU layer only takes in past observations but not past actions. We found that a larger number of hidden units in the BC-RNN model leads to significant overfitting. Both BC-MLP and BC-RNN receive 3 input observations and output probability distributions over 15 discrete actions.

C.2 AIDA Implementation

The AIDA implementation follows the value-iteration network and QMDP network [240, 228] to enable end-to-end training in Pytorch [301]. We used a state dimension of 20, action dimension of 15, and a maximum planning horizon of 30 steps (3 seconds). The Normalizing Flow network consisted of a Gaussian mixture base distribution and a two-layer MLP network with ReLU activation and 30 hidden units in each layer. For each mini-batch of observation-action sequences, we first computed the log likelihood of the observations at all time steps and used (5.4) to compute the belief sequences. We then computed the cumulative EFE in (5.8) and the resulting optimal policy in (5.9) for each inferred belief using the QMDP approximation method [42]. We evaluated dataset action likelihood using a weighted average of optimal policies over different horizons:

$$\pi(a|b) = \sum_{H} \pi(a|b, H)P(H) \tag{C.1}$$

where $P(H)$ is a truncated Poisson distribution up to the maximum planning horizon.

The QMDP method assumes the belief-action value can be approximated as a weighted-average

of the state-action value:

$$\mathcal{G}^*(b_t, a_t) = \sum_{s_t} b_t(s_t)\mathcal{G}^*(s_t, a_t) \tag{C.2}$$

where

$$\mathcal{G}^*(s_t, a_t) = EFE(s_t, a_t) + \log \pi(a_t|b_t) + \sum_{s_{t+1}} P(s_{t+1}|s_t, a_t)\mathcal{G}^*(s_{t+1}) \tag{C.3}$$

$$EFE(s_t, a_t) = D_{KL}(P(s_{t+1}|s_t, a_t)||\tilde{P}(s_{t+1})) + \mathbb{E}_{P(s_{t+1}|s_t,a_t)}[\mathcal{H}(P(o_{t+1}|s_{t+1}))] \tag{C.4}$$

and $\forall s \in \mathcal{S}, \mathcal{G}^*(s_{t+H+1}) = 0$.

The combination of QMDP approximation and computing the observation entropy in (C.4) using the Gaussian base distributions reduced the model's ability to evaluate state uncertainty. However, given the low state uncertainty shown in Figure 5.7 and Figure 5.8 (i.e., the nearly deterministic belief states in the lower right charts), these approximations do not significantly impact the current results while providing the benefit of computational tractability.

Another difference between our implementation and the common active inference presentation is that we performed exact Bayesian state inference (i.e., (5.4)) instead of approximate variational inference (e.g., in [18]). This does not impact the current results since both methods arrive at the same solution in the discrete state setting.

C.3 AIDA-MPC Implementation

In the AIDA-MPC model, we replaced the learned discrete environment dynamics model with a physics-based dynamics model with deterministic state transition and observation functions. The physical-based model had the same three observation modalities: $d, \Delta v, \tau^{-1}$. We defined the state space as $\{d, \Delta v\}$. Assuming constant lead vehicle acceleration, the state transition function is the

following linear function:

$$\begin{bmatrix} d' \\ \Delta v' \end{bmatrix} = \begin{bmatrix} 1 & -\delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} d \\ \Delta v \end{bmatrix} + \begin{bmatrix} 0 \\ -\delta t \end{bmatrix} a \tag{C.5}$$

The state to observation mapping for d and Δv are identity functions. The observation τ^{-1} is computed as $\tau^{-1} \approx \Delta v/d$.

Given this dynamics model, we used the Cross-Entropy Method (CEM [238]) model predictive controller to generate actions treating the AIDA log preference probability over observation as the reward function, i.e., $R(o) = \log \sum_s P(o|s)\tilde{P}(s)$. At each time step, the CEM controller is initialized with a Gaussian distribution over finite horizon action sequences. It then iteratively refines the distribution by sampling N action sequences from the distribution, simulating the action sequences forward using the dynamics model, refitting the Gaussian distribution to the top K samples. Finally, it selects the first step of the mean action sequence of the final Gaussian distribution as the action output. We used a CEM planning horizon of 6 time steps (0.6 seconds), sampled $N = 50$ action sequences, selected the top $K = 5$ sequences, and refined the distribution for 20 iterations.

C.4 Parameter Counts

The number of parameters in each model is listed in Table C.1.

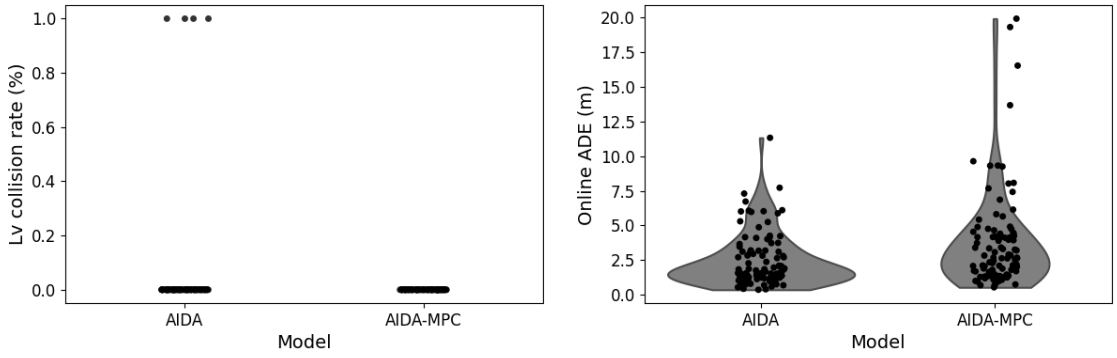
Table C.1: Parameter count of all models.

	IDM	BC-MLP	BC-RNN	AIDA
Count	6	4125	6465	7670

C.5 AIDA vs. AIDA-MPC

Figure C.1a and C.1b show AIDA and AIDA-MPC's lead vehicle collision rate and ADE for each tested trajectory, respectively, where each point corresponds to the result of a trajectory. The

shadows in Figure C.1b represent the density of each model's ADEs, where wider shadow represents higher density.



(a) (b)

Figure C.1: Online same-lane evaluation results of AIDA and AIDA-MPC. Each point represents a trajectory in the test set. The AIDA-MPC replaces the AIDA's dynamics model with a physics-based dynamics model and plans by treating the AIDA's preference distribution as a reward function using model-predictive control. (a) Lead vehicle collision rate of each trajectory. (b) ADE of each trajectory. Wider shadows represent higher density of the ADE values.

APPENDIX D

APPENDIX FOR CHAPTER 6

D.1 Proofs for section 6.4

D.1.1 Proofs for Section 6.4.1

Derivation of BTOM Gradients (section 6.4.1). Recall the definition of the optimal entropy-regularized policy and value functions:

$$\begin{aligned} \hat{\pi}(a|s;\theta) &= \frac{\exp(Q_\theta(s,a))}{\sum_{\tilde{a}} \exp(Q_\theta(s,\tilde{a}))} \\ Q_\theta(s,a) &= R_{\theta_1}(s,a) + \gamma \mathbb{E}_{\hat{P}_{\theta_2}(\cdot|s,a)}[V_\theta(s')] \\ V_\theta(s) &= \log \sum_{\tilde{a}} \exp(Q_\theta(s,\tilde{a})) \end{aligned} \tag{D.1}$$

The gradient of the policy log likelihood in terms of the Q function gradient is obtained as follow:

$$\begin{aligned} \nabla_\theta \log \hat{\pi}(a|s;\theta) &= \nabla_\theta Q_\theta(s,a) - \nabla_\theta V_\theta(s) \\ &= \nabla_\theta Q_\theta(s,a) - \frac{1}{Z_\theta} \nabla_\theta \sum_{\tilde{a}} \exp(Q_\theta(s,\tilde{a})) \\ &= \nabla_\theta Q_\theta(s,a) - \frac{1}{Z_\theta} \sum_{\tilde{a}} \exp(\nabla_\theta Q_\theta(s,\tilde{a})) \\ &= \nabla_\theta Q_\theta(s,a) - \mathbb{E}_{\tilde{a}\sim\hat{\pi}(\cdot|s;\theta)}[\nabla_\theta Q_\theta(s,\tilde{a})] \end{aligned} \tag{D.2}$$

where $Z_\theta = \sum_{a'} \exp(Q_\theta(s,a'))$ is the normalizer.

Recall $\rho^{\hat{\pi}}_{\hat{P}}(\tilde{s},\tilde{a}|s,a)$ is the discounted state-action occupancy measure starting from pair (s,a). We define for any function $f(s,a)$:

$$\mathbb{E}_{\rho^{\hat{\pi}}_{\hat{P}}(\tilde{s},\tilde{a}|s,a)}[f(s,a)] = \mathbb{E}_{\tau\sim(\hat{P},\hat{\pi})}\left[\sum_{t=0}^{\infty} \gamma^t f(s,a) \middle| s_0 = s, a_0 = a\right] \tag{D.3}$$

We now derive Q function gradients with respect to the reward parameters θ_1 and dynamics

parameters θ_2, respectively.

$$
\begin{aligned}
\nabla_{\theta_1} Q_\theta(s,a) &= \nabla_{\theta_1} R_{\theta_1}(s,a) + \gamma \mathbb{E}_{s' \sim \hat{P}_{\theta_2}(\cdot|s,a)}[\nabla_{\theta_1} V_\theta(s')] \\
&= \nabla_{\theta_1} R_{\theta_1}(s,a) + \gamma \mathbb{E}_{s' \sim \hat{P}_{\theta_2}(\cdot|s,a), a' \sim \hat{\pi}(\cdot|s';\theta)}[\nabla_{\theta_1} Q_\theta(s',a')] \\
&= \nabla_{\theta_1} R_{\theta_1}(s,a) + \gamma \mathbb{E}_{s' \sim \hat{P}_{\theta_2}(\cdot|s,a), a' \sim \hat{\pi}(\cdot|s';\theta)} \Big[\\
&\quad \nabla_{\theta_1} R_{\theta_1}(s',a') + \gamma \mathbb{E}_{s'' \sim \hat{P}_{\theta_2}(\cdot|s',a'), a'' \sim \hat{\pi}(\cdot|s'';\theta)}[\nabla_{\theta_1} Q_\theta(s'',a'')] \Big] \\
&= \nabla_{\theta_1} R_{\theta_1}(s,a) + \mathbb{E}_{\tau \sim (\hat{P}, \hat{\pi})} \left[\sum_{h=1}^{\infty} \gamma^h \nabla_{\theta_1} R_{\theta_1}(s_h, a_h) \middle| s_0 = s, a_0 = a \right] \\
&= \mathbb{E}_{\rho^{\hat{\pi}}_{\hat{P}}(\tilde{s},\tilde{a}|s,a)} \left[\nabla_{\theta_1} R_{\theta_1}(\tilde{s},\tilde{a}) \right]
\end{aligned}
\tag{D.4}
$$

In line two we used the result that $\nabla_\phi V_\phi(s)$ for both $\phi = \theta_1$ and $\phi = \theta_2$ corresponds to the second term in (D.2) .

$$
\begin{aligned}
\nabla_{\theta_2} Q_\theta(s,a) &= \nabla_{\theta_2} R_{\theta_1}(s,a) + \nabla_{\theta_2} \gamma \mathbb{E}_{s' \sim \hat{P}_{\theta_2}(\cdot|s,a)}[V_\theta(s')] \\
&= \gamma \sum_{\tilde{s}} V_\theta(\tilde{s}) \nabla_{\theta_2} \hat{P}_{\theta_2}(\tilde{s}|s,a) + \gamma \mathbb{E}_{s' \sim \hat{P}_{\theta_2}(\cdot|s,a), a' \sim \hat{\pi}(\cdot|s';\theta)}[\nabla_{\theta_2} Q_\theta(s',a')] \\
&= \gamma \sum_{\tilde{s}} V_\theta(\tilde{s}) \nabla_{\theta_2} \hat{P}_{\theta_2}(\tilde{s}|s,a) + \gamma \mathbb{E}_{s' \sim \hat{P}_{\theta_2}(\cdot|s,a), a' \sim \hat{\pi}(\cdot|s';\theta)} \Big[\\
&\quad \gamma \sum_{\tilde{s}} V_\theta(\tilde{s}) \nabla_{\theta_2} \hat{P}_{\theta_2}(\tilde{s}|s',a') + \gamma \mathbb{E}_{s'' \sim \hat{P}_{\theta_2}(\cdot|s',a'), a'' \sim \hat{\pi}(\cdot|s'';\theta)}[\nabla_{\theta_2} Q_\theta(s'',a'')] \Big] \\
&= \gamma \sum_{\tilde{s}} V_\theta(\tilde{s}) \nabla_{\theta_2} \hat{P}_{\theta_2}(\tilde{s}|s,a) + \mathbb{E}_{\tau \sim (\hat{P}, \hat{\pi})} \left[\sum_{h=1}^{\infty} \gamma^{h+1} \sum_{\tilde{s}} V_\theta(\tilde{s}) \nabla_{\theta_2} \hat{P}_{\theta_2}(\tilde{s}|s_h,a_h) \middle| s_0 = s, a_0 = a \right] \\
&= \mathbb{E}_{\rho^{\hat{\pi}}_{\hat{P}}(\tilde{s},\tilde{a}|s,a)} \left[\gamma \sum_{s'} V_\theta(s') \nabla_{\theta_2} \hat{P}_{\theta_2}(s'|\tilde{s},\tilde{a}) \right]
\end{aligned}
\tag{D.5}
$$

We make a quick remark on the identifiability of simultaneous estimation.

Remark 4. *Simultaneous reward-dynamics estimation of the form (6.5) without specific assumptions on the prior $P(\theta)$ is in general unidentifiable.*

Proof. Let $\mathbf{R} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ and $\mathbf{P} \in \mathbb{R}_{+}^{|\mathcal{S}||\mathcal{A}|\times|\mathcal{S}|}, \sum_{s'} \mathbf{P}^{a}_{ss'} = 1$, $\mathbf{Q} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ and $\mathbf{V} \in \mathbb{R}^{|\mathcal{S}|}$ be a set of Bellman-consistent reward, dynamics, and value functions in matrix form. Let $\mathbf{P'} \neq \mathbf{P}$ be an alternative dynamics model. We can always find an alternative reward $\mathbf{R}' = \mathbf{R} + \Delta\mathbf{R}$, where:

$$\begin{aligned} \Delta\mathbf{R} &= (\mathbf{Q} - \mathbf{Q}) - \gamma(\mathbf{P}'\mathbf{V} - \mathbf{P}\mathbf{V}) \\ &= -\gamma\Delta\mathbf{P}\mathbf{V} \end{aligned} \tag{D.6}$$

without changing the value functions and optimal entropy-regularized policy. □

Remark 4 implies that existing simultaneous estimation approaches which do not use explicit or implicit regularizations, such as the SERD algorithm by [124], cannot in general accurately estimate expert reward. Paired with theorem 3, it shows that these algorithms cannot in general achieve good performance.

D.1.2 Proofs for Section 6.4.2

Derivation of discounted likelihood (6.14).

$$
\begin{aligned}
&\mathbb{E}_{P(\tau)}\left[\sum_{t=0}^{\infty}\gamma^t \log \hat{\pi}_\theta(a_t|s_t)\right] \\
&= \mathbb{E}_{P(\tau)}\left[\sum_{t=0}^{\infty}\gamma^t \left(Q_\theta(s_t,a_t) - V_\theta(s_t)\right)\right] \\
&= \mathbb{E}_{P(\tau)}\left[\sum_{t=0}^{\infty}\gamma^t \left(R_{\theta_1}(s_t,a_t) + \gamma \mathbb{E}_{s'\sim\hat{P}_{\theta_2}(\cdot|s_t,a_t)}[V_\theta(s')]\right)\right] - \mathbb{E}_{P(\tau)}\left[\sum_{t=0}^{\infty}\gamma^t V_\theta(s_t)\right] \\
&= \mathbb{E}_{P(\tau)}\left[\sum_{t=0}^{\infty}\gamma^t R_{\theta_1}(s_t,a_t)\right] - \mathbb{E}_{\mu}\Big[V_\theta(s_0)\Big] \\
&\quad + \mathbb{E}_{P(\tau)}\left[\sum_{t=0}^{\infty}\gamma^{t+1}\mathbb{E}_{s'\sim\hat{P}_\theta(\cdot|s_t,a_t)}[V_\theta(s')]\right] - \mathbb{E}_{P(\tau)}\left[\sum_{t=1}^{\infty}\gamma^t V_\theta(s_t)\right] \\
&= \mathbb{E}_{P(\tau)}\left[\sum_{t=0}^{\infty}\gamma^t R_{\theta_1}(s_t,a_t)\right] - \mathbb{E}_{\mu}\Big[V_\theta(s_0)\Big] \\
&\quad + \mathbb{E}_{P(\tau)}\left[\sum_{t=0}^{\infty}\gamma^{t+1}\mathbb{E}_{s'\sim\hat{P}_\theta(\cdot|s_t,a_t)}[V_\theta(s')]\right] - \mathbb{E}_{P(\tau)}\left[\sum_{t=0}^{\infty}\gamma^{t+1}\mathbb{E}_{s'\sim P(\cdot|s_t,a_t)}[V_\theta(s')]\right] \\
&= \underbrace{\mathbb{E}_{\rho^\pi_P}\Big[R_{\theta_1}(s_t,a_t)\Big] - \mathbb{E}_{\mu}\Big[V_\theta(s_0)\Big]}_{\ell(\theta)} + \underbrace{\gamma\mathbb{E}_{\rho^\pi_P}\Big[\mathbb{E}_{s'\sim\hat{P}_\theta(\cdot|s_t,a_t)}V_\theta(s') - \mathbb{E}_{s''\sim P(\cdot|s_t,a_t)}V_\theta(s'')\Big]}_{\textbf{T1}}
\end{aligned}
\tag{D.7}
$$

The following lemma shows that **T1** is negligible if the estimated dynamics is accurate under the *expert* distribution, which is available from the offline dataset.

Lemma 5. *(Restate of lemma 1) Let* $\epsilon = \mathbb{E}_{(s,a)\sim P(\tau)} D_{KL}(P(\cdot|s,a)||\hat{P}(\cdot|s,a))$ *be the dynamics estimation error and* $R_{max} = \max_{s,a}|R_\theta(s,a)| + \log|\mathcal{A}|$ *be an upper bound on reward and policy entropy, it holds that*

$$
|\textbf{\textit{T1}}| \leq \frac{\gamma R_{max}}{(1-\gamma)^2}\sqrt{2\epsilon} \tag{D.8}
$$

Proof.

$$
\begin{aligned}
|\mathbf{T1}| &= \left| \sum_{t=0}^{\infty} \gamma^{t+1} \mathbb{E}_{(s_t,a_t)\sim P(\tau)} \left[\sum_{s'} V_\theta(s') \left(\hat{P}(s'|s_t,a_t) - P(s'|s_t,a_t) \right) \right] \right| \\
&\overset{(1)}{\leq} \sum_{t=0}^{\infty} \gamma^{t+1} \mathbb{E}_{(s_t,a_t)\sim P(\tau)} \left[\sum_{s'} |V_\theta(s')| \left| \hat{P}(s'|s_t,a_t) - P(s'|s_t,a_t) \right| \right] \\
&\overset{(2)}{\leq} \sum_{t=0}^{\infty} \gamma^{t+1} \|V_\theta(\cdot)\|_\infty \mathbb{E}_{(s_t,a_t)\sim P(\tau)} \left[\left\| \hat{P}(\cdot|s_t,a_t) - P(\cdot|s_t,a_t) \right\|_1 \right] \\
&\overset{(3)}{\leq} \sum_{t=0}^{\infty} \gamma^{t+1} \|V_\theta(\cdot)\|_\infty \sqrt{2\mathbb{E}_{(s_t,a_t)\sim P(\tau)} D_{KL}(P||\hat{P})} \\
&= \frac{\gamma}{1-\gamma} \|V_\theta(\cdot)\|_\infty \sqrt{2\epsilon}
\end{aligned}
$$

where (1) follows from Jensen's inequality, (2) follows from Holder's inequality, and (3) follows from Pinsker's inequality.

Finally, given $\mathcal{H}(\pi(a|s)) = -\sum_a \pi(a|s) \log \pi(a|s) \leq -\sum_a \pi(a|s) \log \frac{1}{|\mathcal{A}|} = \log |\mathcal{A}|$, we have $\|V_\theta(\cdot)\|_\infty \leq \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \left(\max_{s,a} |R_\theta(s,a)| + \log |\mathcal{A}|\right)\right] = \frac{R_{\max}}{1-\gamma}$.

□

D.1.3 Proofs for Section 6.4.4

Theorem 3 uses the results from [273] (restated in lemma 2), which decomposes the real environment performance gap between the expert and the learner into their policy and model advantages in the estimated dynamics.

Theorem 6. *(Restate of theorem 3) Let $\epsilon_{\hat{\pi}} = -\mathbb{E}_{(s,a)\sim d_P^{\pi}}[\log \hat{\pi}_{\hat{P}}(a|s)]$ be the policy estimation error and $\epsilon_{\hat{P}} = \mathbb{E}_{(s,a)\sim d_P^{\pi}} D_{KL}[P(\cdot|s,a)||\hat{P}(\cdot|s,a)]$ be the dynamics estimation error. Assuming bounded expert-learner marginal state-action density ratio $\left\| \frac{d_P^{\hat{\pi}}(s,a)}{d_P^{\pi}(s,a)} \right\|_\infty \leq C$, we have the following (absolute) performance bound for the IRL agent:*

$$
|J_P(\hat{\pi}) - J_P(\pi)| \leq \frac{1}{1-\gamma} \epsilon_{\hat{\pi}} + \frac{\gamma(C+1)R_{max}}{(1-\gamma)^2} \sqrt{2\epsilon_{\hat{P}}} \tag{D.9}
$$

Proof.

$$
\begin{aligned}
&|J_P(\hat{\pi}) - J_P(\pi)| \\
&\leq \frac{1}{1-\gamma}\epsilon_{\hat{\pi}} \\
&\quad + \frac{\gamma}{1-\gamma}\mathbb{E}_{(s,a)\sim d_P^{\pi}}\left[\left|\frac{d_P^{\hat{\pi}}(s,a)}{d_P^{\pi}(s,a)}\left(\mathbb{E}_{s'\sim P}V_{\hat{P}}^{\hat{\pi}}(s') - \mathbb{E}_{s''\sim \hat{P}}V_{\hat{P}}^{\hat{\pi}}(s'')\right)\right|\right] \\
&\quad + \frac{\gamma}{1-\gamma}\mathbb{E}_{(s,a)\sim d_P^{\pi}}\left[\left|\mathbb{E}_{s'\sim \hat{P}}V_{\hat{P}}^{\hat{\pi}}(s') - \mathbb{E}_{s''\sim P}V_{\hat{P}}^{\hat{\pi}}(s'')\right|\right] \\
&\leq \frac{1}{1-\gamma}\epsilon_{\hat{\pi}} \\
&\quad + \frac{\gamma}{1-\gamma}\left\|\frac{d_P^{\hat{\pi}}(\cdot,\cdot)}{d_P^{\pi}(\cdot,\cdot)}\right\|_{\infty}\left\|V_{\hat{P}}^{\hat{\pi}}(\cdot)\right\|_{\infty}\mathbb{E}_{(s,a)\sim d_P^{\pi}}\left[\left\|\hat{P}(\cdot|s,a) - P(\cdot|s,a)\right\|_1\right] \\
&\quad + \frac{\gamma}{1-\gamma}\left\|V_{\hat{P}}^{\hat{\pi}}(\cdot)\right\|_{\infty}\mathbb{E}_{(s,a)\sim d_P^{\pi}}\left[\left\|\hat{P}(\cdot|s,a) - P(\cdot|s,a)\right\|_1\right] \\
&= \frac{1}{1-\gamma}\epsilon_{\hat{\pi}} + \frac{\gamma(C+1)R_{\max}}{(1-\gamma)^2}\sqrt{2\epsilon_{\hat{P}}}
\end{aligned}
\tag{D.10}
$$

where the last line uses results from lemma 1. □

D.2 Implementation Details

Our implementation builds on top of the official RAMBO implementation[1] [267].

D.2.1 MuJoCo Benchmarks

For the MuJoCo benchmarks described in section 6.5.2, we follow standard practices in model-based RL.

D.2.1.1 Dynamics Pre-training

We use an ensemble of $K = 7$ neural networks where each network outputs the mean and covariance parameters of a Gaussian distribution over the difference between the next state and the current state $\delta = s' - s$:

$$
\hat{P}_{\theta_2}^{(k)}(\delta|s,a) = \mathcal{N}(\delta|\mu_{\theta_2}^{(k)}(s,a), \Sigma_{\theta_2}^{(k)}(s,a)) \tag{D.11}
$$

[1] `https://github.com/marc-rigter/rambo`

Each network is a 4-layer feedforward network with 200 hidden units and Sigmoid linear unit (SiLU) activation function. For the initial pre-training step, we maximize the likelihood of dataset transitions using a batch size of 256 and early stop when all models stop improving for more than 1 percent. We then select the 5 best models in terms of mean-squared-error on a 10 % holdout validation set. During model rollouts, we randomly pick one of the 5 best models (elites) to sample the next state.

Table D.1: Shared hyperparameters across different environments

	Hyparameter	BTOM	RTOM
SAC + MBPO	critic learning rate	3e-4	3e-4
	actor learning rate	3e-4	3e-4
	discount factor (γ)	0.99	0.99
	soft target update parameter (τ)	5e-3	5e-3
	target entropy	-dim(A)	-dim(A)
	minimum temperature (α)	0.1	0.001
	batch size	256	256
	real ratio	0.5	0.5
	model retain epochs	5	5
	training epochs	500	300
	steps per epoch	1000	1000
Dynamics	# model networks	7	7
	# elites	5	5
	adv. rollout batch size	1000	256
	adv. rollout steps	10	10
	adv. update steps	50	50
	adv. loss weighting (λ_1)	0.01	0.01
	supervised. loss weighting (λ_2)	1	1
	learning rate	1e-4	1e-4
	adv. update steps	50	50
Reward	max reward	10	10
	rollout batch size	1000	64
	rollout steps	40	100
	l2 penalty	1e-3	1e-3
	learning rate	1e-4	1e-4
	update steps	1	1

Table D.2: Environment-specific hyperparameters

Environment	Hyperparameter	BTOM
Hopper	model rollout batch size	10000
	model rollout steps	40
	model rollout frequency	250
HalfCheetah	model rollout batch size	50000
	model rollout steps	5
	model rollout frequency	250
Walker2d	model rollout batch size	10000
	model rollout steps	40
	model rollout frequency	250

D.2.1.2 Policy Training

Our policy training process follows MBPO [47] which uses SAC with automatic temperature tuning [302]. Shared hyperparameters across different environments are listed in Table D.1 and environment-specific hyperparameters are listed in Table D.2. For the actor and critic, we use feedforward neural networks with 2 hidden layers of 256 units and ReLU activation. We train the actor and critic networks using a combination of real and simulated samples. We use a real ratio of 0.5, which is standard practice in model-based RL and IRL. We found that BTOM requires a higher minimum temperature to stablize training, which is set to $\alpha = 0.1$.

We found that different MuJoCo environments require different model rollout hyperparameters, similar to what's reported in [263]. Specifically, Hopper and Walker2d only work with significantly larger rollout steps. We decrease their rollout batch size to reduce computational overhead. HalfCheetah on the other hand works better with smaller rollout steps and larger rollout batch size. In contrast to Lu et al. (2021), we did not use different rollout hyperparameters for different datasets.

D.2.1.3 Reward and Dynamics Training

We use 10 random trajectories from the D4RL MuJoCo expert dataset after removing all expert trajectories that resulted in terminal states.

We use the same network architecture as the actor-critic to parameterize the reward function. We further clip the reward function to a maximum range of ± 10 and apply l2 regularization on all weights and biases with a penalty of 0.001.

As described in the main text, we update the reward function by simulating sample trajectories and taking a single gradient step. For RTOM, we randomly sample expert trajectory segments of length “rollout steps" and use the first step as the start of our simulated sample paths.

We update the dynamics using on-policy rollouts branched from the dataset state-actions. We use the same batch size for reward and dynamics rollouts, which is 1000 for BTOM and 256 for RTOM. Because only the first step in BTOM sample paths comes from the dataset, it requires a larger batch size to iterate more data samples. We also train BTOM for more epochs than RTOM.

To compute the dynamics log likelihood in the REINFORCE gradient in (6.18), we treat the ensemble as a uniform mixture and compute the likelihood as:

$$\hat{P}_{\theta_2}(\delta|s,a) = \frac{1}{K}\sum_{k=1}^{K} \hat{P}_{\theta_2}^{(k)}(\delta|s,a) \tag{D.12}$$

We set the dynamics adversarial loss weighting to $\lambda_1 = 0.01$ for both BTOM and RTOM. We found this to work better than what's in the official RAMBO implementation, which is $\lambda_1 = 0.0768$. Note that the RAMBO author reported $\lambda_1 =$ 3e-4 in their paper but forget to average their REINFORCE loss over the mini-batch of size 256 in their implementation, which is instead treated as a sum by default by TensorFlow. We empirically found that small λ_1 leads to severe model exploitation.