THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Robustness During Learning, Interaction and Adaptation for Autonomous Driving

Hannes Eriksson

Department of Computer Science and Engineering
Chalmers University of Technology
Gothenburg, Sweden, 2023

Robustness During Learning, Interaction and Adaptation for Autonomous Driving

Hannes Eriksson

ISBN 978-91-7905-904-0
Doktorsavhandlingar vid Chalmers tekniska högskola, Ny series nr 5370'
ISSN 0346-718X
Department of Computer Science and Engineering
Chalmers University of Technology
SE-412 96 Gothenburg, Sweden
Phone: +46 (0)31 772 1000
www.chalmers.se

Printed by Chalmers Reproservice
Gothenburg, Sweden, August 2023

Abstract

In a sequential decision-making process, it is imperative to consider the potential risks of taking incorrect decisions throughout the whole process as all wrongdoings may not be possible to be remedied. This is particularly important when there are potentially catastrophic consequences. In this work, we develop robust decision-making processes, doing appropriate risk assessments where needed, to be able to plan to avoid unacceptable consequences. In contrast to traditional techniques for decision-making under uncertainty that aim to maximise performance in expectation, we choose to value other aspects out of the distribution of outcomes. For instance, in an application such as autonomous driving, the chance of causing an accident might be small yet fatal. A risk-averse decision-maker may choose to modify the risk criterion to only include consider e.g. the 25% worst-case outcomes to design a more robust decision-making process. We propose frameworks for quantifying uncertainty under the reinforcement learning framework and develop robust algorithms and theory that allow for risk-sensitive decision-making under uncertainty. Further, we study the interactions between multiple agents in autonomous systems and ways to deploy decision-making processes to novel scenarios by adaptation.

Keywords: Reinforcement learning, autonomous driving, risk-sensitive learning, uncertainty estimation.

List of Publications

This thesis is based on the following publications:

[A] **Hannes Eriksson**, Christos Dimitrakakis, "Epistemic Risk-Sensitive Reinforcement Learning". Published in The 28th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), 2020.

[B] Emilio Jorge, **Hannes Eriksson**, Christos Dimitrakakis, Debabrota Basu, Divya Grover, "Inferential Induction: A Novel Framework for Bayesian Reinforcement Learning". Published in Proceedings of Machine Learning Research Volume 137 (PMLR), 2021.

[C] **Hannes Eriksson**, Debabrota Basu, Mina Alibeigi, Christos Dimitrakakis, "SENTINEL: Taming Uncertainty with Ensemble-based Distributional Reinforcement Learning". Published in The 38th Conference on Uncertainty in Artificial Intelligence (UAI), 2022.

[D] **Hannes Eriksson**, Debabrota Basu, Mina Alibeigi, Christos Dimitrakakis, "Risk-Sensitive Bayesian Games for Multi-Agent Reinforcement Learning under Policy Uncertainty". Accepted for The 13th Workshop on Optimization and Learning in Multiagent Systems @ AAMAS'22.

[E] Thomas Kleine Buening, Christos Dimitrakakis, **Hannes Eriksson**, Divya Grover, Emilio Jorge, "Minimax-Bayes Reinforcement Learning". Published in The 26th International Conference on Artificial Intelligence and Statistics (AISTATS), 2023.

[F] **Hannes Eriksson**, Debabrota Basu, Tommy Tram, Mina Alibeigi, Christos Dimitrakakis, "Reinforcement Learning in the Wild with Maximum Likelihood-based Model Transfer". To be submitted to Transactions on Intelligent Transportation Systems, 2023.

Other publications by the author, not included in this thesis, are:

[G] **Hannes Eriksson**, Christos Dimitrakakis, Lars Carlsson, "High-Dimensional Near-Optimal Experiment Design for Drug Discovery via Bayesian Sparse Sampling". *arXiv preprint arXiv:2104.11834, 2021.*

[H] Emilio Jorge, **Hannes Eriksson**, Christos Dimitrakakis, Debabrota Basu, Divya Grover, "On Bayesian Value Function Distributions". *The 15th European Workshop on Reinforcement Learning, 2022.*

Acknowledgments

I want to thank all the people in the Wallenberg AI, Autonomous Systems and Software Program (WASP) for all the interesting times we have had during travels, courses and conferences. Likewise, I would like to thank the people I have interacted with at Chalmers University of Technology during my stay here. I would also like to thank my colleagues at Zenseact AB for all the interesting times and challenges related to our work in autonomous driving. I would like to give special thanks to the people at Harvard SEAS who were involved with my stay there before the inception of my PhD studies, these were, David, Paul, Goran, Rafael and more, who inspired me to work on my PhD. I would also like to give extra thanks to the people involved with the advanced graduate program at Zenseact AB, for fostering a great community for research related to autonomous driving, these are Mats, Carl and more.

I would also like to thank my supervisors, these are, Alexander, Nasser, who helped guide me in the early parts of this PhD work, Mina, who has been guiding me for the latter parts of the projects and has shown great commitment and interest in our work. I would like to give special thanks to Devdatt, who initially suggested that I apply for a PhD position, for being an examiner both for my master's thesis and PhD thesis and for his role in the Data Science division at Chalmers University of Technology. Furthermore, I would like to thank the people part of my licentiate and PhD committees, Marc, Aviv, Philippe, Niaho, Emilie, Morteza and Fredrik.

I would like to give thanks to the members of the research group under Christos, which now span multiple countries, these are (former included) Aristide, Divya, Emilio, and Debabrota, who I have had the pleasure of working with and discussing these years, as well as Thomas, Meirav and Ann-Marie in our weekly discussions. Finally, I would like to thank Christos who has endured all these years with me, from my master's thesis work to my work as a project assistant, to my research trip to Boston and throughout my PhD studies. Without his assistance and guidance, this would never have been possible.

Contents

Part I

Overview

CHAPTER 1

Introduction

Designing an *autonomous agent*, that is, an agent that can act independently, without external input, to be able to act in a real-world scenario is a challenging task. The agent must be able to interact safely with other beings, agents and objects as well as adapt to newly learned experiences obtained from exploring the environment.

The agent has to adhere to a sequential decision-making process, that is, it iteratively chooses which actions to take, records how the actions affect the world and based on this, modifies its action selection procedure.

Instead of directly deploying the agent in the real-world one may attempt to create a *closed-loop system* with the agent inside of it, e.g., a system that describes how the system changes without any need for external human inputs. In this system, the agent can learn and interact with other agents without putting other people in danger. In this work, we study sequential decision-making problems in closed-loop systems. An obvious issue is the possible mismatch between the simulator and the real world, however, we hope these findings can help inform real-world decision-making processes.

The main framework of note studied in this thesis is the *Reinforcement Learning* (RL) framework which involves unknown closed-loop systems, that

is, the system itself has to be learned online. We note that sometimes these systems will be used interchangeably with the terms model and environment. Such a system includes the reward function or cost function which describes the immediate reward for taking a particular action in a given state. In a sense, this could be viewed as how 'good' was it to take that action without considering what might happen in the future. Another important part of the system is the transition function. This function describes how the system evolves with action inputs from the agent. Since both the reward function and transition function are unknown a priori, this introduces a kind of uncertainty about the system.

We begin by elaborating on the main setting of the sequential decision-making problem studied in this thesis, that is *Autonomous Driving* (AD) and how the mentioned uncertainty manifests and can be handled. Further, we delve into the main framework, RL, and its different forms and how they can be used to guide the design of a robust agent. Finally, we split the thesis into the three main aspects we have looked at, that is, robustness during learning, interaction and adaptation.

1.1 Autonomous Driving

In the autonomous driving setting, we aim to design an autonomous agent capable of driving a vehicle without the assistance of a driver. In particular, it should be able to reliably transport the vehicle from between two locations while adhering to traffic rules and safety norms. There are multiple possible definitions of the complete AD pipeline and the one presented here consists of four modules: *perception*, *prediction*, *planning* and *control*. Perception, which involves taking sensory inputs from e.g. camera images, LIDAR, etc. and combining them into a set of outputs using *sensor fusion*. Conventionally, the final output of this module may be for instance a set of bounding boxes surrounding possible objects in a scenario. Given these object bounding boxes the prediction module's purpose is to determine possible road user trajectories, i.e., how are other vehicles, pedestrians, etc. going to behave in the future. This set of road user trajectories can then be fed into the planning module used to construct a path for the autonomous vehicle, taking into account where other road users may travel to. Finally, the proposed path is used by the control module to control things like the steering and throttle and its goal

is to ensure the vehicle follows the proposed path.

1.2 Uncertainty in Autonomous Driving

In most AD settings there needs to be explicit considerations of inherent and extraneous uncertainties to design a robust agent. For instance, a vehicle can be described using a physical model and this model could be used to make inferences about how the vehicle will move. However, each vehicle may have its associated physical model based on its properties such as weight, engine, etc. From this, it would be reasonable to infer the optimal agent could be different for each possible vehicle. Furthermore, the agent has to take in sensory inputs from the scene such as camera images, LIDAR, GPS, map data, etc. These inputs may also be imperfect and introduce additional uncertainties into the setting. In addition to this, the vehicle needs to interact with other road users where we have to make inferences about how they will act. All these uncertainties compound and the decision-making process needs to take into account that it might learn things about the scenario in the future to make the correct decisions in the present.

1.3 Reinforcement Learning

The main framework of interest that can formulate decision-making problems under uncertainty is the RL framework. This framework has seen great success[1]–[3], and is something that has been studied extensively for the field of AD as well [4], [5]. One of the main features of the RL framework is the construction of a *Markov Decision Process* (MDP) [6], which is a model used to describe how the process evolves under stimulus by a decision-maker. A rigorous formalisation of the MDP will be given in Chapter 2. Typically, the true underlying MDP is unknown and the agent has to estimate this MDP from available data. This introduces a sort of uncertainty related to the knowledge available to the agent, henceforth to be called *epistemic uncertainty* [7]–[10]. This exists in contrast to another kind of uncertainty, which is inherent to the MDP and is termed *aleatory uncertainty* [11]. Aleatory uncertainty is abundant in applications with high stochasticity, such as games of chance. In applications such as autonomous driving, with mostly deterministic mechanics, this source of uncertainty might not be so great, given that world

dynamics are known. These two kinds of uncertainties form the basis of this thesis and the differences, applications and importance of them will be stressed throughout this thesis.

1.4 Research Questions

The main overarching research question we set out to answer throughout this work is **How to Design Autonomous Agents that Drive Safely?** This is an important question to be able to answer for anyone who intends to deploy a live agent into a traffic situation with other road users and objects that we need to be considerate about. The question is very broad and can be decomposed into more approachable research questions. One such research question that has been the main focus to us is **How to Learn Safely?** Here, we are concerned with safety throughout the learning process. Whenever an agent is deployed into an unknown environment it may be tasked with learning new things. We then wish to guarantee the agent does not create excessive risks for himself and others. A similar albeit different research question we considered is **How to Interact Safely?** In this case, we are studying the interactions between the agent and other road users. Explicit care must be given to the preferences and intentions of other road users to ensure safe driving. Finally, we investigated **How to Transfer Knowledge from Known Scenarios to a Novel Scenario?** In this case, we might have an agent that has learned how to drive in Sweden and Germany. Now, we wish to deploy this agent in India. We wish to extract as much knowledge as possible from the known domains while being open to learning important aspects of the novel domain. What considerations do we have to take? Can it be deployed without issue?

1.5 Contributions

We decompose our contributions included in this thesis into three categories. These are **robustness during learning**, where we study epistemic uncertainty. In this case, this relates to model uncertainty where the model is unknown. It can also be viewed as uncertainty due to the lack of data. This field of study is particularly important when an agent is learning a novel task. We may then want to be robust with respect to what we do not yet know. The ultimate goal here is to design a robust learner that reduces risk

(i.e., unacceptable consequences with high probability) and acts more safely. In these works, we look at risk-sensitive reinforcement learning, model-based reinforcement learning and distributional reinforcement learning.

The next field of study is **robustness during interaction**. When interacting with other agents one may have to infer not only what the other agents are doing but what they *know*. A robust agent can take into account how other agents will adapt to your actions. For these lines of work, we mainly consider the game setting. In particular, Bayesian games for studying uncertainty about other agent's behaviour and Minimax formulations of adversarial games.

The last category is **robustness during adaptation** which deals with adapting to novel tasks. This could for instance. In this case, we study concepts such as multi-task reinforcement learning and transfer reinforcement learning.

Robustness during learning. In Eriksson and Dimitrakakis [9] we develop and introduce a risk-sensitive Bayesian RL framework for decision-making under *epistemic* uncertainty for discrete and continuous state space RL problems. In addition to that, we propose two algorithms, one based on approximate dynamic programming and one based on the Bayesian policy gradient.

In the work Jorge *et al.* [12] we introduce a novel framework for Bayesian distributional RL by appropriately marginalising out the variables in such a way that three new approaches can be formulated. We propose one of them, Bayesian Backwards Induction and demonstrate its performance in the paper.

Further, in Eriksson *et al.* [10] we propose a novel risk measure, termed *composite risk*, which takes into account both aleatory and epistemic uncertainty and appropriately weights them together. We prove superiority over previous methods of joining the risk measures theoretically and propose an ensemble-based algorithm that can quantify this new risk measure.

Lastly, in Buening *et al.* [13] we propose a novel framework of Minimax-Bayes RL, whereby the decision-maker is searching for a policy that is robust to changes in belief. We prove that under certain conditions, there exists a minimax solution and we provide two alternative methods of obtaining it.

Robustness during interaction. In Eriksson *et al.* [14] we investigate epistemic uncertainty in the context of Bayesian games. This allows a set of agents

to be risk-sensitive with respect to what they believe other agents will do. We provide a method of smoothly obtaining a joint set of policies for the agents which results in risk-averse behaviours for all the agents.

Robustness during adaptation. In Eriksson *et al.* [15] we study the case of knowledge transfer in RL. In it, we have access to existing knowledge relating to a set of known MDPs. We wish to leverage this knowledge when making decisions in a novel task. The proposed framework demonstrates superiority compared to methods of learning from scratch. In it, we provide an algorithm fulfilling this objective. Further, we provide theoretical bounds in terms of model deviation for a few specific settings.

1.6 Thesis Outline

The thesis is initiated with a chapter covering the main ingredients the included publications are based upon, in Chapter 2. These include the basics of RL and the constructions which allow for risk-averse decision-making in RL. In the next chapter, Chapter 3, we study decision-making during the learning process and how one can be robust concerning what one does not yet know. Here papers the $[A, B, C, E]$ are discussed. In the next chapter, Chapter 4, we discuss uncertainty in situations where one interacts with other agents. In this chapter, the paper $[D]$ is elaborated upon. Finally, in Chapter 5, we consider situations where an agent must use its existing knowledge to adapt its decision-making about a novel scenario. The paper $[F]$ studied here.

CHAPTER 2

Background

In this chapter, we will cover the necessary prerequisites to understand the complete list of works present in this thesis. The main bulk of the theory involves studying the concept of MDPs, and how to construct and plan inside them. This is available in Section 2.1. We also study the case when the MDP itself is unknown in Section 2.2.

2.1 Dynamic Programming

In this section, we go over the fundamentals of using Dynamic Programming (DP) to solve MDPs.

Definition 1 (Markov Decision Process)**:** *A Markov Decision Process μ is a tuple $\mu = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}, \gamma)$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$, the permissible action set, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a reward function indicating the goodness of taking an action a in a particular state s. $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is a transition kernel, describing the evolution of the process from a state s under the influence of action a, resulting in a transition to state s'. Typically this process is stochastic and induces a probability distribution of successor states. Furthermore, γ is a discount factor determining the effective horizon of the*

problem. Finally, the objective in an MDP is typically to maximise the return (or utility) $R = \sum_{t=0}^{\infty} \gamma^t r_t$ *which is the sum of future discounted rewards.*

In addition to the formalism surrounding the MDP itself, we need to introduce a couple of important concepts involving planning in MDPs. The policy π denotes the strategy of the agent. In principle, the policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ gives an action (or a probability measure over a set of actions in $\mathcal{A}$) for every state. These policies come in many forms, from *deterministic Markov* policies Π^{MD}, where there exist an action $a \in \mathcal{A}$ such that $\pi(a_t = a \,|\, s_t) = 1$, to *stochastic Markov* policies Π^{MS}, where the policy returns a probability measure over actions. The aforementioned policies are Markovian and only depend on the current state. Another set of policies which we will henceforth refer to as *adaptive* policies are policies that are non-Markovian. Let $h_t = (s_0, a_0, r_0, s_1, ..., s_t)$ be the *history* of states, actions and rewards observed up until time t. Furthermore, let $\mathcal{H}$ be the set of all possible histories, then, an adaptive policy $\pi : \mathcal{H} \to \Delta(\mathcal{A})$ is a policy outputting an action (or a probability measure over actions) for every possible history.

One of the most important concepts studied in DP is the *value functions*. Their purpose is to represent the expected return or expected utility for a particular policy and MDP given either a state or a state-action pair. This is important to a decision-maker as they can e.g., be used to compare the quality of competing policies. The value functions come in two flavours, the *state value function* and the *state-action value function*.

Definition 2 (State Value Function)**:** *The state value function* $V_\mu^\pi(s)$ *describes the expected utility of being in state* s*, for MDP* μ*, following policy* π*.*

$$V_\mu^\pi(s) = \mathbb{E}_\mu^\pi\Big[\sum_{t=0}^{\infty} \gamma^t r_t \,|\, s_0 = s\Big], \tag{2.1}$$

where $\gamma \in [0, 1)$ *is the discount factor, determining the effective horizon of the problem.*

Definition 3 (State-action Value Function)**:** *The state-action value function* $Q_\mu^\pi(s, a)$ *describes the expected utility of being in state* s*, for MDP* μ*, taking action* a *and then immediately following policy* π*.*

$$Q_\mu^\pi(s, a) = \mathbb{E}_\mu^\pi\Big[\sum_{t=0}^{\infty} \gamma^t r_t \,|\, s_0 = s, a_0 = a\Big], \tag{2.2}$$

where $\gamma \in [0, 1)$ is the discount factor, determining the effective horizon of the problem.

A planner typically wants to identify the optimal policy $\pi^* \triangleq \arg\max_{\pi \in \Pi} V_\mu^\pi$ or the optimal value function $V_\mu^* \triangleq \max_{\pi \in \Pi} V_\mu^\pi$. As we delve into further topics the underlying MDP μ may be unknown. In the case when it is known and given certain assumptions on the MDP the maximum V_μ^* and the maximiser π^* can easily be obtained.

We can now define a function operator, termed the *Bellman operator*, which can be used to compute value functions.

Definition 4 (Bellman Operator)**:** *The Bellman operator, $\mathscr{P}^\pi : V \to V$ is defined as,*

$$\mathscr{P}^\pi V(s) \triangleq \mathbb{E}_\mu^\pi[\mathcal{R}(s, a)] + \mathbb{E}_\mu^\pi\Big[\mathcal{T}(s, a)V(s')\Big]. \tag{2.3}$$

Iteratively applying $\mathscr{P}^\pi$ for all states $s \in \mathcal{S}$ for a particular MDP μ can be used to obtain the value function associated with the policy π, MDP μ and state s. Another operator of interest is the *Bellman optimality operator*.

Definition 5 (Bellman Optimality Operator)**:** *The Bellman optimality operator, $\mathscr{P} : V \to V$ is defined as,*

$$\mathscr{P} V(s) \triangleq \max_{a \in \mathcal{A}} \mathbb{E}_\mu[\mathcal{R}(s, a)] + \mathbb{E}_\mu\Big[\mathcal{T}(s, a)V(s')\Big]. \tag{2.4}$$

These two operators are contraction mappings (cf. Bertsekas [16]) and thus, repeated applications of them will result in convergence to its corresponding value function, i.e., $\lim_{t\to\infty} \mathscr{P}^\pi\big(\ldots(\mathscr{P}^\pi V_0)\big) = V_\mu^\pi$ and $\lim_{t\to\infty} \mathscr{P}\big(\ldots(\mathscr{P} V_0)\big) = V_\mu^*$.
$\underset{\text{t times}}{}$ $\underset{\text{t times}}{}$

2.2 Reinforcement Learning

When the MDP is unknown we may want to estimate it. One such method is *Bayesian Reinforcement Learning* (BRL) [17] whereby a belief is kept over the set of plausible MDPs. In general, BRL approaches fall under the category of *model-based RL*. We contrast this approach with *model-free RL* where we forego the modelling of the underlying MDP and instead place the focus on the value function.

Bayesian Reinforcement Learning

In the BRL framework, we adopt a Bayesian approach to the RL problem. That is, we have a subjective belief over the possible MDPs. This makes sense in the model-based RL framework as the MDP is unknown and the decision-maker has to estimate the MDP parameters or distribution over them from interactions with the environment. The subjective belief over MDPs can come in many forms, depending on the setting. For instance, there may be a finite set of plausible MDPs and the subjective belief β could then represent a probability vector over those MDPs. In many cases, the set of plausible MDPs is infinite. More formally, let $(\mathcal{M}, \mathcal{F}, \beta)$ be a probability space over MDPs with appropriate $\sigma-$algebra. Then, $\beta(\mu) \triangleq \mathbb{P}(\mu)$ is a *prior* probability distribution over MDPs $\mu \in \mathcal{M}$. As the decision-maker acquires experiences from interacting with the environment they would like to update their subjective belief about the MDP. Let $\mathcal{D}_t$ denote the observed data up until time t and let $\mathbb{P}(\mathcal{D}_t \,|\, \mu)$ denote the *likelihood function*, that is, the joint probability of $\mathcal{D}_t$ given the MDP μ. Then, the *posterior* probability distribution $\beta(\mu \,|\, \mathcal{D}_t) \triangleq \mathbb{P}(\mu \,|\, \mathcal{D}_t)$ is the conditional probability of the MDP μ given the observed data $\mathcal{D}_t$, this follows from Bayes' rule.

$$\underset{\text{Posterior}}{\mathbb{P}(\mu \,|\, \mathcal{D}_t)} \propto \underset{\text{Likelihood}}{\mathbb{P}(\mathcal{D}_t \,|\, \mu)} \times \underset{\text{Prior}}{\mathbb{P}(\mu)} \tag{2.5}$$

This framework has some helpful properties for a model-based decision-maker. To start with, the agent can at all times sample an MDP from the posterior and use the aforementioned techniques to arrive at a policy optimal for the sample. An algorithm can be constructed this way and it is commonly termed *Posterior Sampling for Reinforcement Learning* (PSRL) [18]–[20]. This algorithm exhibits numerous interesting qualities such as its simplicity to deploy and its performance [21], [22] (in terms of Bayesian regret). In this work, one of the main algorithms we benchmark against is in fact PSRL.

Given that we now consider distributions over MDPs, the previous formalism surrounding MDPs has to be extended to incorporate this. Let $U(\pi, \mu) \triangleq \mathbb{E}^{\pi}_{\mu}[R]$ be the expected utility for a particular MDP and policy. Then,

$$U(\pi, \beta) \triangleq \int_{\mathcal{M}} U(\pi, \mu)\beta(\mu)\,\mathrm{d}\mu, \tag{2.6}$$

is the expected utility marginalised over the subjective belief β. In BRL our objective is typically to maximise Equation 2.6, also termed the *Bayesian Value Function*. The maximising policy $\pi^* \in \arg\max_\pi \int_{\mathcal{M}} U(\pi, \mu)\beta(\mu)\,\mathrm{d}\mu$ is called the *Bayes-optimal policy* and is for all but the most simple of scenarios, very challenging to identify. In particular, the policy is adaptive. Optimising for adaptive policies can be done in numerous ways, including using history-dependent policies, count policies or tree policies, see Duff [23].

From Equation 2.6 one might also identify the role the belief plays to the decision-maker. β induces a probability distribution over value functions (or expected utility). In Strens [19] and Dearden *et al.* [24] the authors focus on the distribution over the MDP itself but one may also choose to represent the induced value function distribution [25]–[27]. These uncertainties about the MDP or value function are as aforementioned termed epistemic uncertainty and considerable focus is given to it throughout this work.

Some consideration has to be taken for which prior, likelihood and posterior to select. If the underlying MDP is inadmissible under the prior β, then it is likewise inadmissible under the posterior. Throughout this work, we will mainly look at prior posteriors of the following three families, inverse-Normal-Gamma priors [28] and Dirichlet priors for tabular MDPs and Bayesian linear regression priors [29] for continuous settings.

Distributional Reinforcement Learning

Under the standard RL framework, the main objective is to maximise the expected utility. In certain applications, it may be useful to be able to represent the full distribution. Here, we will denote the distribution of the return or utility as the *utility distribution* and the distribution over value functions as the *value function distribution*. Indeed, Bellemare *et al.* [30] and Hessel *et al.* [31] demonstrated state-of-the-art performance in Atari games by explicitly modelling the return distribution using histograms. One might ask oneself why learning the complete distribution would be helpful for a risk-neutral decision-maker. Bellemare *et al.* [30] posits part of the reason for the superior performance can be because in this case, there is a more stable learning target. Other approaches using utility distributions are e.g., Tang and Agrawal [32] where the authors learn a Gaussian distribution of the return. The aforementioned works rely on neural network estimators to construct

the utility distributions. Historically, this line of work can be traced back to Dearden *et al.* [33] and Morimura *et al.* [34], where the authors modelled return distributions explicitly. In this work, we build upon the utility distribution framework set forth by Bellemare *et al.* [30]. Let $Z_\theta(s, a)$ denote the utility distribution at state s for action a with the network parameters θ. Furthermore, let N denote the number of histograms and $V_{\text{MIN}}, V_{\text{MAX}}$ the lower and upper bounds of the histogram distribution respectively, with support $\{z_i = V_{\text{MIN}} + i\Delta z : 0 \leq i < N\}, \Delta z := \frac{V_{\text{MAX}} - V_{\text{MIN}}}{N-1}$. Then, the probability associated with the ith histogram is given by,

$$Z_\theta(s, a) = z_i \quad \text{w.p.} \quad p_i(s, a) := \frac{e^{\theta_i(s,a)}}{\sum_j e^{\theta_j(s,a)}}. \tag{2.7}$$

This formulation allows for simple optimisation of the distribution parameters and because of its discrete nature, easy computable distributional statistics. However, it does come with some drawbacks. Notably, the lower and upper bounds of the utility distribution need to be known a priori and its expressiveness is highly dependent on the number of histograms chosen to make up the distribution.

We will now instead consider the distribution over the value functions themselves. As previously mentioned, this uncertainty may arise when the underlying MDP is unknown. Every MDP μ together with a policy π and starting state s has an associated value function with it, $\mathbb{E}^\pi_\mu[R]$. A decision-maker may want to quantify this uncertainty explicitly as in the case of utility distributions, either to allow for risk-sensitive decision-making [7], [10], [35], [36], optimism in the face of uncertainty [25], [26] or perhaps because it is expected to yield more robust estimators as in the case of utility distributions. Throughout this work, we will focus on two procedures. Firstly, the Bayesian perspective, i.e., a prior-posterior procedure over MDPs is created and the induced value function distribution is computed by sampling models from the prior, evaluating them using the current policy and constructing the distribution. Secondly, a procedure where a neural network, a set of neural networks or a statistical model is used to learn the value function distribution explicitly. In Jorge *et al.* [12] and O'Donoghue *et al.* [26] the uncertainty about the value function itself is modelled. This captures the epistemic uncertainty in a model-free way and allows for efficient exploration by ignoring the aleatory uncertainty.

Risk-Sensitive Reinforcement Learning

The general RL setting is concerned with maximising performance in expectation, i.e. $\pi^* \in \arg\max_\pi \mathbb{E}^\pi_\mu[R]$. However, in applications such as autonomous driving, it may be more interesting to consider a modified objective. For instance, the naive objective may be *minimise travel time between A and B* but the decision-maker may also want to limit the probability of accidents occurring. One may do so by adding a high penalty on near-accidents or one could define a surrogate objective. For instance, *maximise performance in the p% worst-case of outcomes*. This is the approach of [10], [37], [38] where a conditional value-at-risk (CVaR) [39] objective is optimised for instead of the expected return. One field of research tasked with this is the *Risk-Sensitive Reinforcement Learning* (RSRL) field. In particular, in our research, we mainly focus on the RSRL setting studying epistemic risk. Epistemic RSRL is concerned with the uncertainty that arises due to the lack of knowledge of the MDP or the data. This has several connections with the BRL setting as the uncertainty about the MDP induces a probability distribution over value functions. This is studied in e.g., [9], [10], [40], [41].

Multi-Task Reinforcement Learning

Sometimes there may not be a unique underlying MDP but a distribution of possible MDPs. An agent may wish to optimise performance given this distribution. This fits neatly into the BRL framework and the maximiser of Equation 2.6 maximises the performance for the MDPs in $\mathcal{M}$ given the probability distribution β.

In some cases, one may have access to a simulator [42] and want to abstract from it to a novel task. This problem setting is commonly referred to as the *transfer reinforcement learning* setting. Here, one may choose to transfer knowledge about the policy [43], value function [44] or as in the case of Eriksson *et al.* [15], via model transfer. In this case, we have a set of *source* MDPs $\mathcal{M}_s \triangleq \{\mu_i\}_{i=1}^m$ and a *target* MDP μ^*. The objective is to identify a policy maximising performance in the target task while making use of existing knowledge about $\mathcal{M}_s$.

CHAPTER 3

Robustness During Learning

The risk I took was calculated, but man, am I bad at math.
—Mincing Mockingbird, *2011*

In this chapter, we investigate robustness during learning. That is, there exists a learning problem. For instance, we aim to deploy an autonomous vehicle in a novel environment. The vehicle needs to be able to interact with other agents in the environment, both static and dynamic, collect experiences and update its driving behaviour while minimising excessive risks to itself, other agents in the environment and the environment itself. This is a difficult task as *the agent does not know what it does not know!* As such, it needs to balance a trade-off between exploration, that is, trying out new actions, visiting new locations, etc., and doing what it currently knows is the best to do, also known as exploitation. So, we set out to answer the following question, **How can we ensure safety during learning?**

To answer this question, we set out to investigate topics concerned with epistemic uncertainty, model uncertainty and the uncertainty due to the lack of data. The ultimate goal of a robust learner is to learn *safely*, that is, learn without taking excessive risk. Such a learner may choose to explore more

conservatively to avoid catastrophic events during learning. One significant drawback of this framework is, however, as we limit the rate of exploration we may in certain cases, increase the time it takes to learn the optimal behaviour. Further, in some specific cases, this limitation may be so strong that it *never* can identify the optimal way of acting.

Our contributions to this field of study are the following.

Value Function Distributional RL. We constructed a novel framework able to handle the value function distribution induced by the model uncertainty. We make a key insight that the value function and model can not be decoupled without significant assumptions. In this work, my contributions lie with the construction of the experimental code base, the prior, part of the algorithm in the continuous case and writing.

Epistemic Risk-Sensitive RL. We developed a framework allowing for risk-sensitive decision-making in the face of epistemic uncertainty. My contributions are with the theory, code base and writing.

Unifying Aleatory and Composite Risk. We developed a framework able to unify aleatory and composite risk into a single risk measure. We demonstrate it inherits properties from the two risk measures. My contributions lie in construction of the framework, code base and writing.

Minimax Bayesian RL. We developed a framework able to consider worst-case distributions over MDPs for agents. We construct a game between the agent, who selects strategies and the environment, who selects distributions over MDPs. The design admits for the agent to obtain a policy with worst-case Bayesian regret guarantees. My contributions to this work lie in experimental and algorithm design of the infinite MDP setting.

Altogether, our contributions aid the design of agents conscious of epistemic uncertainty. This is by allowing for uncertainty quantification, theoretical justifications, worst-case analysis and decision-making using estimators considering epistemic uncertainty. We will now delve into individual contributions.

3.1 Induced Value Function Distributions

Here we are interested in the induced uncertainty about the value functions induced by the uncertainty about the MDPs. This is of interest since it has been shown in previous chapters that modelling the full distribution rather than just the expectation may lead to a more robust learning process as well as

in some cases, superior performance. Furthermore, modelling the full distribution admits us to use of optimism in the face of uncertainty and risk-sensitive decision-making. However, the main focus of this work is with respect to risk-neutral performance.

As mentioned in Section 2.2, introducing a probability space $(\mathcal{M}, \mathcal{F}, \beta)$ over MDPs $\mu \in \mathcal{M}$ will lead to a distribution over value functions, with each MDP being associated with its value function. In general, we have the following dependency on the value function distribution on the belief β, policy π and the data $\mathcal{D}$,

$$\mathbb{P}^{\pi}_{\beta}(V \mid \mathcal{D}) = \int_{\mathcal{M}} \mathbb{P}^{\pi}_{\mu}(V) \, \mathrm{d}\beta(\mu \mid \mathcal{D}). \tag{3.1}$$

To get a feel of what Equation 3.1 says. The inner term $\mathbb{P}^{\pi}_{\mu})(V)$ is the prior over value functions, conditioned on μ and π. It is known that for a given MDP and policy pair it uniquely defines a value function. The term we marginalise over, $\beta(\mu \mid \mathcal{D})$, is the posterior over μ given the data $\mathcal{D}$. These together give us the complete value function distribution conditioned on the data. As we can see from this equation, the value function distribution depends on the posterior and the policy.

Now that the distribution has been defined a decision-maker can choose whether it wants to update its policy in the direction that maximises the expectation of this quantity or whether it wants to design an optimistic or risk-sensitive agent by optimising instead for the tail expectation or similar of this distribution.

3.2 Risk-Sensitive Reinforcement Learning with Exponential Utilities

In this case, we focus on the design of a risk-sensitive agent in the face of epistemic uncertainty during learning. This is of particular interest for risk-averse decision-makers since if an agent is to be deployed in an unknown environment it needs to learn safely. This is also challenging as if the agent is too conservative, it may not explore nearly enough to identify the optimal behaviour. Crucially, an epistemic risk-sensitive agent will, with perfect knowledge of the underlying MDP, converge to the optimal behaviour of the risk-neutral agent. This is because in that case, there is no epistemic uncertainty. In certain

scenarios, such as for multi-task settings, there may still exist epistemic uncertainty even with perfect knowledge of the underlying MDPs, as there is a distribution over MDPs. We begin by deciding on an appropriate utility function for this problem.

Let the utility function be $U(x) = \frac{1}{\alpha} \log \mathbb{E}\Big[\exp \alpha x \Big]$, motivated initially by Mihatsch and Neuneier [11]. In this case, $\alpha \in \mathbb{R}$ is a parameter controlling the risk-sensitiveness of the utility function. Let $R = \sum_{t=0}^{\infty} \gamma^t r_t$ and $(\mathcal{M}, \mathcal{F}, \beta)$ be a probability space over MDPs. Then, this yields the following objective,

$$\nabla_\theta U(\pi, \beta) = \nabla_\theta \frac{1}{\alpha} \log \int_{\mathcal{M}} e^{\alpha \mathbb{E}^\pi_\mu[R]} \, \mathrm{d}\beta(\mu) \tag{3.2}$$

$$= \frac{\int_{\mathcal{M}} \nabla_\theta e^{\alpha \mathbb{E}^\pi_\mu[R]} \, \mathrm{d}\beta(\mu)}{\alpha \int_{\mathcal{M}} e^{\alpha \mathbb{E}^\pi_\mu[R]} \, \mathrm{d}\beta(\mu)} \tag{3.3}$$

$$= \frac{\alpha \int_{\mathcal{M}} e^{\alpha \mathbb{E}^\pi_\mu[R]} \nabla_\theta \mathbb{E}^\pi_\mu[R] \, \mathrm{d}\beta(\mu)}{\alpha \int_{\mathcal{M}} e^{\alpha \mathbb{E}^\pi_\mu[R]} \, \mathrm{d}\beta(\mu)} \tag{3.4}$$

$$= \frac{\int_{\mathcal{M}} e^{\alpha \mathbb{E}^\pi_\mu[R]} \nabla_\theta \mathbb{E}^\pi_\mu[R] \, \mathrm{d}\beta(\mu)}{\int_{\mathcal{M}} e^{\alpha \mathbb{E}^\pi_\mu[R]} \, \mathrm{d}\beta(\mu)}. \tag{3.5}$$

Estimates for $\mathbb{E}^\pi_\mu[R]$ can be obtained by using rollouts in MDP μ using policy π. When doing this, it is pertinent to use different rollouts for the three estimated quantities. The objective in Equation 3.2 is quite similar to existing works [45], [46], although they are taking the integral over actions instead of models.

An algorithm optimising for this objective can be seen in Algorithm 1.

We also provide an algorithm based on approximate dynamic programming, following the work of Dimitrakakis [47], in Algorithm 2. We leave the experimental results section to the paper in Eriksson and Dimitrakakis [9].

3.3 Decision-Making under Composite Risk Measures

There is a breadth of existing works studying aleatory risk [7], [11], [48], [49], epistemic risk [7], [9], [35] and joint risk [35], [50]. In this work, we also aimed to unify the two risks into a joint risk measure. Furthermore, we wanted to do

Algorithm 1 Epistemic Risk Sensitive Policy Gradient (ERSPG)

Input: Policy parametrisation θ_t, β_t (current posterior).
repeat
  `Simulate to get` θ_{t+1}
  for $i = 1$ **to** N **do**
    $\mu_{(1)}, \mu_{(2)} \sim \beta_t$
    for $j = 1$ **to** M **do**
      $\tau^{(1)}_{\mu_{(1)}}, \tau^{(2)}_{\mu_{(1)}} \sim \pi_\theta, \mu_{(1)}$
      $\tau^{(3)}_{\mu_{(2)}} \sim \pi_\theta, \mu_{(2)}$
    end for
  end for
  $\theta_{t+1} \leftarrow \theta_t - \Big[\sum_{i=0}^{N} \exp\Big(\alpha {\tau_{\mu_i}}^{(1)}\Big) {\tau_{\mu_i}}^{(2)} \nabla_\theta \log \pi_\theta(a|s)] / [\sum_{i=0}^{N} \exp\Big(\alpha {\tau_{\mu_i}}^{(3)}\Big)\Big]$

  Deploy $\pi_{\theta_{t+1}}$ and obtain $\tau \sim \mu, \pi_{\theta_{t+1}}$
  $\xi_{t+1} \leftarrow \beta_t, \tau$
until *convergence*

it in a rigorous manner using risk measures. We accomplish this by composing the two risk measures. We also show that for the final risk measure to exhibit similar properties as its two constituents, it needs to be of special construction. For instance, other works such as [35], [50] will not work as the variance is not a *coherent* risk measure.

A risk measure $U : \mathcal{X} \to \mathbb{R}$ is a function from a probability distribution to a scalar. This construction allows decision-makers to compare risks under different distributions and choose what best adheres to their risk profile. One class of risk measures that has garnered a lot of interest recently is the coherent risk measures, given by Artzner *et al.* [51]. According to the definition, a coherent risk measure $U : \mathcal{X} \to \mathbb{R}$ has to satisfy four axioms:

Axiom 1 (Monotonicity)**:** *If* $X \leq Y$ *almost surely,* $U(X) \leq U(Y)$.

Axiom 2 (Positive homogeneity)**:** *For any* $c \geq 0$, $U(cX) = cU(X)$.

Axiom 3 (Translation invariance)**:** *For any constant* $a \in \mathbb{R}$, $U(X + a) = U(X) + a$.

Axiom 4 (Subadditivity)**:** *For* $X, Y \in \mathcal{X}$, $U(X + Y) \leq U(X) + U(Y)$.

In the work Eriksson *et al.* [10] our focus is on risk measures of this kind.

Algorithm 2 Epistemic Risk Sensitive Backwards Induction (ERSBI)

Input: $\mathcal{M}$ (set of MDPs), β (current posterior)
repeat
  for $\mu \in \mathcal{M}$ $s \in \mathcal{S}$, $a \in \mathcal{A}$ **do**
    $Q_\mu(s,a) = \mathcal{R}_\mu(s,a) + \gamma \sum_{s'} \mathcal{T}_\mu^{ss'} V_\mu(s')$
  end for
  for $s \in \mathcal{S}, a \in \mathcal{A}$ **do**
    $\mathcal{Q}_\beta(s,a) = \sum_\mu \xi(\mu) U[(Q_\mu(s,a)]$
  end for
  $\pi(s) = \arg\max_a \mathcal{Q}_\beta(s,a)$.
  for $\mu \in \mathcal{M}$ **do**
    $V_\mu(s) = Q_\mu(s, \pi(s))$.
  end for
until *convergence*
return π

Quantifying Composite Risk Measures

Following Eriksson *et al.* [10] we define the risk measures of interest. To start with, we define the risk of the random variable Z under the distorted utility function U_α in three different ways for clarity.

$$\begin{aligned}\mathrm{Risk}_{U_\alpha}(Z) &\triangleq \int_{\mathcal{Z}} Z \,\mathrm{d}(U_\alpha \circ P) \\ &= \int_{\mathcal{Z}} U_\alpha(1 - F_Z(z)) \,\mathrm{d}z = \int_0^1 U_\alpha(t) \,\mathrm{d}q(1-t). \end{aligned} \tag{3.6}$$

Moving on with the risk measure associated with aleatory uncertainty, that is the uncertainty that arises due to the inherent stochasticity of the MDP μ and policy π, we chose the following definition.

Aleatory Risk. Given a coherent risk measure with distorted utility function U_α^A, the aleatory risk is quantified as the deviation of the total risk of

individual models from the risk of the average model.

$$A(U_\alpha^A, \beta) \triangleq \int_\Theta \int_\mathcal{Z} Z \, \mathrm{d}(U_\alpha^A \circ \mathbb{P})(Z|\theta) \, \mathrm{d}\beta(\theta) - \int_\Theta \int_\mathcal{Z} \hat{Z} \, \mathrm{d}(U_\alpha^A \circ \mathbb{P})(\hat{Z})$$

Epistemic Risk. Given a coherent risk measure with distorted utility function U_α^E, the epistemic risk quantifies the uncertainty invoked by not knowing the true model. Thus, the risk can be computed over any statistics of the models, such as the expectation.

$$E(U_\alpha^E, \beta) \triangleq \int_\Theta \int_\mathcal{Z} Z \, \mathrm{d}\mathbb{P}(Z|\theta) \, \mathrm{d}(U_\alpha^E \circ \beta)(\theta)$$

Composite Risk under Model and Inherent Uncertainty. Finally, in [10] a joint risk measure termed composite risk is defined that takes into account both the uncertainty that arises due to the true MDP μ being unknown, as well as the MDPs are inherently stochastic. The total uncertainty is then a combination of both these sources of uncertainty and in order to quantify the total uncertainty, we proposed *composite risk*.

Definition 6 (Composite Risk)**:** *For two coherent risk measures with distorted utility functions $U_{\alpha_1}^A$ and $U_{\alpha_2}^E$, belief distribution β on model parameters $\theta \in \Theta$, and a random variable $Z \in \mathcal{Z}$, the composite risk of epistemic and aleatory uncertainties is defined as*

$$\begin{aligned} F^C(U_{\alpha_1}^A, U_{\alpha_2}^E, \beta) &\triangleq \mathrm{Risk}_{U_{\alpha_2}^E}(\mathrm{Risk}_{U_{\alpha_1}^A}(Z|\theta)|\beta) \\ &= \int_\Theta \int_\mathcal{Z} Z \, \mathrm{d}(U_{\alpha_1}^A \circ \mathbb{P})(Z|\theta) \, \mathrm{d}(U_{\alpha_2}^E \circ \beta)(\theta) \\ &= \int_0^1 \int_0^1 U_{\alpha_2}^E(v) U_{\alpha_1}^A(u) \, \mathrm{d}q_{Z|\theta}(1-u) \, \mathrm{d}q_\beta(1-v) \end{aligned} \tag{3.7}$$

The inclusion of a composite risk measure allows for a more accurate representation of the total uncertainty compared to existing works optimising jointly over both risks, such as in [7], [35].

Theorem 5 (Coherence)**:** *If $U_{\alpha_1}^A$ and $U_{\alpha_2}^E$ are distorted utilities for two coherent risk measures, the composite risk measure $F^C(U_{\alpha_1}^A, U_{\alpha_2}^E, \beta)$ is also coherent.*

The theorem Theorem 5 is important so as to retain coherency after composing the risk measures.

Theorem 6: *We are given two sources of aleatory and epistemic uncertainties ξ_1 and ξ_2. If $U^A_{\alpha_1}$ and $U^E_{\alpha_2}$ are distortion measures for two coherent risk measures quantifying aleatory and epistemic risks respectively, then, i) $F^A(U^A_{\alpha_1}, \beta) = F^C(U^A_{\alpha_1}, I, \beta)$, where I is the identity function, and ii) $F^C(U^A_{\alpha_1}, U^E_{\alpha_2}, \beta) \geq F^A(U^A_{\alpha_1}, \beta)$, if $\alpha_2 \neq 1$.*

This theorem is used in the work Eriksson *et al.* [10] to demonstrate the superiority of the composed risk measure approach to an additive risk approach to jointly optimising for both risks. The proofs of the theorems Theorem 5 and Theorem 6 are left for the interested reader in the paper Eriksson *et al.* [10].

In our work, we propose an algorithm for optimising composite risk measures as defined in Eq. 3.7. The full algorithm is available in Algorithm 3.

3.4 Minimax Robustness in the Face of Model Uncertainty

If one were to deploy an agent into an environment and wanted it to act safely, one method would be to identify what would be the *worst-case* environment and then find the best-performing policy in that case. This would give us a guaranteed worst-case performance, as in all other environments it would perform at least as good or better. In this work, we are interested in the case where the MDP is unknown. This line of thinking has spurred numerous works, such as Mannor *et al.* [52] and Wiesemann *et al.* [53], where they construct uncertainty sets around the MDP parameters and identify policies with worst-case performance w.r.t. those uncertainty sets. Our approach here is to construct a robust policy using a *minimax formulation*. In it, a policy is optimised against an adversary selecting for the worst-case environment or distribution over environments. Our approach here is the latter.

Bayesian Minimax Theorems

In Buening *et al.* [13] we instead chose to study minimax theorems for distributions over MDPs. Let $R(\pi, \mu) \triangleq U(\pi^*, \mu) - U(\pi, \mu)$ be the *regret* associated with a policy π for a particular MDP μ. The regret measures how far off the

Algorithm 3 SENTINEL-K with Composite Risk

Input: Initial state s_0, action set $\mathcal{A}$, distortion measures $U^A_{\alpha_1}, U^E_{\alpha_2}$, hyperparameter λ, target networks $[\theta_1^-, ..., \theta_K^-]$, value networks $[\theta_1, ..., \theta_K]$, update schedule Γ_1, Γ_2.
for $t = 1, 2, \ldots$ **do**
  //* Update K-value and target networks for estimating return distributions *//
  for $t' \in \Gamma_1 \cup \Gamma_2$ **do**
    Generate $\{D_1, ..., D_K\} \leftarrow \text{DataMask}(\mathcal{D}^{t'})$
    for $i = 1, \ldots, K$ **do**
      Sample mini batch $\tau \sim D_i$
      $F^C(Z(s_t, a)|U^A_{\alpha_1}, U^E_{\alpha_2}, \beta)$ using τ and K-target networks $\{\theta_i^-\}_{i=1}^K$.
      Get $a^* = \arg\max_a F^C(Z(s_t, a)|U^A_{\alpha_1}, U^E_{\alpha_2}, \beta)$
      Update value network θ_i using τ, a^*
      Update target network θ_i^- using τ, a^* if $t' \in \Gamma_1$
    end for
  end for
  //* Estimate the composite risk of each action using the estimated return distributions *//
  for $a \in \mathcal{A}$ **do**
    Compute weights $\mathbf{w} = w_1, ..., w_K$.
    for i **in** K **do**
      Compute aleatory risks $Q_i^A(s_t, a)$ from $\int_{\mathcal{Z}} Z \, \mathrm{d}(U^A_{\alpha_1} \circ \mathbb{P})(Z|\theta_i)$
    end for
    Compute composite risk over weighted aleatory estimates $Q^C(s_t, a) = \text{Risk}_{U^E_{\alpha_2}}(\{w_i Q_i^A(s_t, a)\}_{i=1}^K)$
  end for
  //* Action selection *//
  Take action $a_t = \arg\max_a Q^C(s_t, a)$
  Observe s_t and update the dataset $\mathcal{D}^t \leftarrow \mathcal{D}^{t-1} \cup \{s_t, a_{t-1}, s_{t-1}, r_{t-1}\}$
end for

optimal policy π^* the current policy π is. We further introduce the notion of *Bayesian regret*, $L(\pi, \beta) \triangleq \int_{\mathcal{M}} R(\pi, \mu) \, \mathrm{d}\beta(\mu)$ which is the average regret of the policy π, marginalised over the belief μ. In particular, in Buening *et al.* [13] we show that using the Bayesian regret as the objective for the zero-sum game, under some conditions, has a value,

$$\min_{\pi \in \Pi} \max_{\beta} L(\pi, \beta) = \max_{\beta} \min_{\pi \in \Pi} L(\pi, \beta). \tag{3.8}$$

This allows us to construct two algorithms that can be used to identify the minimax policy and maximin prior. The minimax theorem in Equation 3.8 only holds in certain cases. For instance, if $\max_{\pi \in \Pi} L(\pi, \beta)$ is convex with respect to β and differentiable everywhere. In practice, this limits the theory to simple settings such as the finite MDP setting. Nevertheless, in our work, we also demonstrate similar aspects in settings beyond what the theory requires. Next, we describe a procedure based on gradient descent ascent that can be used to obtain the solution.

Computing Minimax Bayesian Regret Gradients

Let $U(\tau) \triangleq \sum_{(s_t, a_t, r_t) \sim \tau_t} \gamma^t r_t$ be the utility associated with a rollout τ, then, the utility of a policy evaluated on an MDP can be estimated by $U(\pi, \mu) = \mathbb{E}_{\tau \sim \pi, \mu}\big[U(\tau)\big]$. This gives us a method of computing the necessarily utilities for the policy gradient procedure. Let us first investigate the case of the update of the agent's policy parameters.

$$\begin{aligned}
\nabla_\pi L(\pi, \beta) &= \nabla_\pi \int_{\mathcal{M}} R(\pi, \mu) \, \mathrm{d}\beta(\mu) && (3.9)\\
&= \nabla_\pi \int_{\mathcal{M}} \Big[U(\pi^*, \mu) - U(\pi, \mu)\Big] \, \mathrm{d}\beta(\mu) && (3.10)\\
&= \nabla_\pi \int_{\mathcal{M}} U(\pi^*, \mu) \, \mathrm{d}\beta(\mu) - \nabla_\pi \int_{\mathcal{M}} U(\pi, \mu) \, \mathrm{d}\beta(\mu) && (3.11)\\
&= -\nabla_\pi \int_{\mathcal{M}} U(\pi, \mu) \, \mathrm{d}\beta(\mu) && (3.12)\\
&= -\int_{\mathcal{M}} \nabla_\pi \, \mathbb{E}_{\tau \sim \pi, \mu}\big[U(\tau)\big] \, \mathrm{d}\beta(\mu). && (3.13)
\end{aligned}$$

Thus, for the policy, optimisation is quite simple. In the case of the prior, it becomes slightly more contrived.

$$\nabla_\beta L(\pi, \beta) = \nabla_\beta \int_{\mathcal{M}} R(\pi, \mu)\, \mathrm{d}\beta(\mu) \tag{3.14}$$

$$= \nabla_\beta \int_{\mathcal{M}} \Big[U(\pi^*, \mu) - U(\pi, \mu)\Big]\, \mathrm{d}\beta(\mu) \tag{3.15}$$

$$= \int_{\mathcal{M}} \nabla_\beta \Big[\mathbb{E}_{\tau\sim\pi^*,\mu}\big[U(\tau)\big] - \mathbb{E}_{\tau\sim\pi,\mu}\big[U(\tau)\big]\Big]\, \mathrm{d}\beta(\mu) \tag{3.16}$$

$$= \int_{\mathcal{M}} \Big[\nabla_\beta\, \mathbb{E}_{\tau\sim\pi^*,\mu}\big[U(\tau)\big] - \nabla_\beta\, \mathbb{E}_{\tau\sim\pi,\mu}\big[U(\tau)\big]\Big]\, \mathrm{d}\beta(\mu). \tag{3.17}$$

In this case, we clearly have a dependence not only on trajectories obtained from the agent's policy but also from the optimal policy. In practice, this means one would have to do separate rollouts using the optimal policy and the agent's policy in order to estimate the utility.

There are several well-known variations of estimating the utility gradient. For instance, one may simply use the sum of rewards, add a baseline or use the reward-to-go formulation [54]. For a rigorous overview, see Schulman *et al.* [55]. In our work, we use a reward-to-go formulation akin to REINFORCE in order to reduce the variance. Further, we subtract a baseline to improve it even further.

Having computed the gradients, we can iteratively update the parameters of our policy and prior,

$$\pi_{t+1} \leftarrow \pi_t - \eta_\pi \nabla_\pi L(\pi, \beta) \tag{3.18}$$

$$\beta_{t+1} \leftarrow \beta_t + \eta_\beta \nabla_\beta L(\pi, \beta). \tag{3.19}$$

In the general case, the obtained solution after iterating this procedure will only be approximately minimax. If, however, $L(\pi, \beta)$ is convex with respect to β and differentiable everywhere, then an exact minimax solution can be found.

CHAPTER 4

Robustness During Interaction

The best way to predict the future is to invent it.

—Alan Kay, *1971*

In this chapter, we sought to identify ways to ensure safety when interacting with other agents. In particular, we are considering interactions with *learning* or *adaptive* agents, that is, agents that may update their behaviour over time. It is well-known that if there are a finite number of agents and the other agents are non-adaptive, the final problem can be reduced to a standard RL problem. Special care needs to be given to problems of this kind as they exist in the intersection between RL and *game theory*. Here, we may choose to study agents of different types. For instance, agents may cooperate towards a common goal, they might be selfish and only attempt to fulfil their own goals or they may sabotage each other. Our objective was to set out to design a framework that can handle all of these, as well as risk-sensitive formulations of them.

Our contributions to this field of study are the following.

Risk-Sensitive Bayesian Games. We constructed a novel framework being able to handle risk-sensitiveness in the Bayesian Games setting, be-

ing able to trade off risks due to player-type uncertainty. In this work, my contributions are with the design, theory, code base and writing.

4.1 Interaction with Stationary Agents

In the case where all the other agents are non-adaptive, the objective is to find the optimal policy first agent, keeping all other policies fixed. This can be modelled as a multi-task RL problem where each set of combinations of policies determines a MDP and the probability of that MDP is the joint probability of all the policies. Thus, this setting can be solved using traditional RL techniques, such as BRL, by constructing a set of possible MDPs and marginalising over them.

4.2 Interaction with Learning Agents

In this setting, we follow the work of Eriksson *et al.* [14]. We start by introducing the concept of a *Bayesian game*, let $\mathcal{G} = (N, K, \mathcal{S}, \mathcal{A}^N, \mathcal{R}^{N\times K}, \mathcal{T}, \beta, \gamma)$ be a game with N players and K possible types. Each distinct player and type combination has its own utility function $\mathcal{R}_i^j$ and $\mathcal{T} : \mathcal{S} \times \mathcal{A}^N \to \Delta(\mathcal{S})$ is the transition distribution associated with the game. Finally, let β be a common prior over types and γ a common discount factor.

In the definition of the game $\mathcal{G}$ there are K^N different policy combinations. We can evaluate the utility of the game by keeping all the policies fixed. For simplicity, assume there are two players and player 1 has type j and player 2 has type k, then, the utility of the first player is,

$$U_1^{j,k}(\mathcal{G}) \triangleq \mathbb{E}_{\mathcal{G}}^{\pi_1^j, \pi_2^k}\big[\sum_{t=0}^{\infty} \gamma^t r_{1,t}^j \,|\, s_0 = s\big]. \tag{4.1}$$

Next, we wish to define the expected utility of a game taking the common prior into account.

Definition 7 (Expected Utility of a game $\mathcal{G}$)**:** *The expected utility marginalised*

over the prior β for a given game $\mathcal{G}$ is

$$\begin{aligned} U_1, U_2 &\triangleq \mathbb{E}_\beta \Big[U_1^{j,k}(\mathcal{G}), U_2^{j,k}(\mathcal{G}) \Big] \\ &= \sum_{j=1}^{K} \sum_{k=1}^{K} \beta(\tau_1 = j, \tau_2 = k)(U_1^{j,k}(\mathcal{G}), U_2^{j,k}(\mathcal{G})), \end{aligned}$$

where τ indicates the type of the player.

We now have a way of computing the utility associated with each player. Next, we introduce the concept of risk in Bayesian games as the uncertainty about types. That is, β, combined with the utility of each of the individual players for that particular type configuration. $U_1^{,,}$ and $U_2^{,,}$ are thus discrete probability distributions with mass equal to $\xi(\tau_1 = j, \tau_2 = k)$ and values $U_1^{j,k}$ and $U_2^{j,k}$. By considering all possible combinations of types we get the following probability mass function,

$$p_{U_1}(U) = \begin{cases} U_1^{1,1}, & \xi(\tau_1 = 1, \tau_2 = 1) \\ U_1^{1,2}, & \xi(\tau_1 = 1, \tau_2 = 2) \\ & \vdots \\ U_1^{K,K}, & \xi(\tau_1 = K, \tau_2 = K), \end{cases} \tag{4.2}$$

and similarly for the other agent, $p_{U_2}(U)$. We wish to construct agents that take the uncertainty about types into account. In particular, the agents should be able to be risk-sensitive w.r.t. this uncertainty. One such objective that admits a lot of flexibility and interpretability is the CVaR objective, focusing on the $\alpha\%$ worst-case outcomes of the distribution. CVaR can be defined as follows, $CVaR_\alpha(U) \triangleq \mathbb{E}[U \,|\, U \leq \nu_\alpha \wedge \mathbb{P}(U \geq \nu_a) = 1 - \alpha]$.

Finally, we investigate three possible techniques to be used to update the policies. The first is *Iterated Best Response* (IBR) [56]. In this case, all the policies are iterated over, one by one, keeping all other policies fixed. Since only a subset of the full parameter set is updated at every step it may result in cycles. The next technique is called *Fictitious Play* (FP) [57]. Here, the agents are still updated iteratively, one by one, however, now they are evaluated using rolling averages of each others' policies. In this case, we will only have smooth policy updates after every full iteration. Lastly, the technique of main focus is a *Dual Ascent Policy Gradient* (DAPG) method, updating all

Algorithm 4 Risk-Sensitive Iterated Best Response/Fictitious Play (RS-IBR/FP)

input : Game $\mathcal{G}$, learning rates η_1, η_2, risk measure ρ_α
for $i = 0 \dots$ convergence **do**
  for $t = 0 \dots$ convergence **do**
    $\theta_{1,i}^{t+1} = \theta_{1,i}^{t} + \eta_1 \nabla_{\theta_1} \left[\rho_\alpha(U_1) \,|\, \mathcal{G}, \theta_{1,i}^{t}, \theta_{2,i}\right]$
  end for
  $\theta_{1,i+1} = \theta_{1,i}^{t}$
  for $t = 0 \dots$ convergence **do**
    if IBR **then**
      $\theta_{2,i}^{t+1} = \theta_{2,i}^{t} + \eta_2 \nabla_{\theta_2} \left[\rho_\alpha(U_2) \,|\, \mathcal{G}, \theta_{1,i+1}, \theta_{2,i}^{t}\right]$
    end if
    if FP **then**
      $\theta_{2,i}^{t+1} = \theta_{2,i}^{t} + \eta_2 \nabla_{\theta_2} \left[\rho_\alpha(U_2) \,|\, \mathcal{G}, \bar{\theta}_{1,i+1}, \theta_{2,i}^{t}\right]$
    end if
  end for
  $\theta_{2,i+1} = \theta_{2,i}^{t}$
end for

Algorithm 5 Risk-Sensitive Dual Ascent Policy Gradient (RS-DAPG)

input : Game $\mathcal{G}$, learning rates η_1, η_2, risk measure ρ_α
for $i = 0 \dots$ convergence **do**
  for $j = 1 \dots K$ **do**
    $\theta_{1,i+1}^{j} = \theta_{1,i}^{j} + \eta_1 \nabla_{\theta_1^j} \Big(\left[\rho_\alpha(U_1) \,|\, \mathcal{G}, \theta_i\right] + \left[\rho_\alpha(U_2) \,|\, \mathcal{G}, \theta_i\right] \Big)$
    $\theta_{2,i+1}^{j} = \theta_{2,i}^{j} + \eta_2 \nabla_{\theta_2^j} \Big(\left[\rho_\alpha(U_1) \,|\, \mathcal{G}, \theta_i\right] + \left[\rho_\alpha(U_2) \,|\, \mathcal{G}, \theta_i\right] \Big)$
  end for
end for

policies simultaneously. The three techniques can be seen in Algorithm 4 and Algorithm 5.

CHAPTER 5

Robustness During Adaptation

All models are wrong, some are useful.

—George Box, *1976*

In many instances, one may want to leverage existing knowledge when faced with a novel task. For such scenarios, one may choose to adopt the transfer reinforcement learning framework. This allows an agent to take experience with similar tasks into account when deployed in a new environment. For instance, given access to a simulator, one may wish to deploy an agent without having to learn from scratch. This may result in the agent being able to utilize a baseline policy with 'good enough' performance or result in more quickly identifying the optimal policy. In Langley [58] and Lazaric [59] the authors describe three main objectives the transfer RL aims to tackle over traditional RL. These are, (i) **learning speed improvement**, i.e., decreasing the amount of data required to learn the solution, (ii) **asymptotic improvement**, where the solution results in better asymptotic performance and (iii) **jumpstart improvement**, where the initial policy results in a better starting solution than the one utilising no previous knowledge.

Our contributions to this field of study are the following.

Model-Based Transfer Reinforcement Learning. We constructed a novel framework being able to handle model transfer to novel domains of the transition function. Earlier works mainly focus on the transfer of reward function, value function or policy. In this work, my contributions are with the design, theory, code base and writing.

5.1 Model-Based Transfer Reinforcement Learning

In Eriksson *et al.* [15] we investigate model-based transfer reinforcement learning, whereby a set of existing source MDPs $\mathcal{M}_s \triangleq \{\mu_i\}_{i=1}^m$ are used to inform decisions in a target MDP μ^*. From here, we can further categorise this into three problem settings. The first is when $\mu^* \in \mathcal{M}_s$. We call this the *I. Finite and Realisable Plausible Models* setting. In this case, identification of the maximum likelihood model is straightforward,

$$\hat{\mu} \in \underset{\mu' \in \mathcal{M}_s}{\arg\max} \log \mathbb{P}(D_t \,|\, \mu'), D_t \sim \mu^*, \tag{5.1}$$

where $\hat{\mu}$ is the maximum likelihood estimator of μ^*. In general, the novel task is not part of the existing set of source models. Consider for example a scenario where an agent has access to m distinct simulators and wants to deploy the agent in the real world. If the real world does not perfectly align with one of the simulators, then we are in the following two settings, the next of which we call the *II. Infinite and Realisable Plausible Models* setting. In this case, we begin by defining the convex set of source MDPs $\mathcal{C}(\mathcal{M}_s) \triangleq \{\mu_1 w_1 + \ldots + \mu_m w_m \,|\, \mu_i \in \mathcal{M}_s, w_i \geq 0, i = 1, \ldots, m, \sum_{i=1}^m w_i = 1\}$. The corresponding optimisation problem is now to find the maximum likelihood model in the mixture of the source MDPs,

$$\hat{\mu} \in \underset{\mu' \in \mathcal{C}(\mathcal{M}_s)}{\arg\max} \log \mathbb{P}(D_t \,|\, \mu'), D_t \sim \mu^*. \tag{5.2}$$

This procedure allows us to not only find the maximum likelihood source MDP as in setting I, but if the target MDP μ^* can be written as a convex combination of the source MDPs, then it is part of the admissible set of MDPs. While we have increased the set of admissible MDPs, this comes at the cost of making the optimisation problem slightly more challenging. Nevertheless, if the number of source MDPs is small then the identification of the maximum

likelihood MDP is rather quick.

Finally, if $\mu^* \notin \mathcal{C}(\mathcal{M}_s)$ the procedure in II has no chance of identifying the true MDP. This setting, we call the *III. Infinite and Non-realisable Plausible Models* setting. In it, it may be possible to find a good proxy model, given μ^* is not too dissimilar from the maximum likelihood estimator $\hat{\mu}$. In our work, we show how the total model deviation depends on the *realisability gap*, which is precisely the gap between the best proxy model $\mu \in \mathcal{C}(\mathcal{M}_s)$ and the true model μ^*, as given by $\epsilon_{\text{Realise}} \triangleq \min_{\mu \in \mathcal{C}(\mathcal{M}_s)} \|\mu^* - \mu\|_1$. Further, let $\epsilon_{\text{Estim}} \|\mu - \hat{\mu}\|_1$. We can now introduce the first theorem of the paper [15],

Theorem 7 (Performance Gap for Non-Realisable Models)**:** *Let $\mu^* = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}^*, \gamma)$ be the true underlying MDP. Further, let $\mu = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}, \gamma)$ be the maximum likelihood $\mu \in \arg\min_{\mu' \in \mathcal{C}(\mathcal{M}_s)} \mathbb{P}(D_\infty \,|\, \mu'), D_\infty \sim \mu^*$ and $\hat{\mu} = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \hat{\mathcal{T}}, \gamma)$ be a maximum likelihood estimator of μ. In addition, let $\pi^*, \pi, \hat{\pi}$ be the optimal policies for the respective MDPs. Then, if $\mathcal{R}$ is a bounded reward function $\forall_{(s,a)} r(s,a) \in [0,1]$ and with ϵ_{Estim} being the estimation error and $\epsilon_{\text{Realise}} \triangleq \min_{\mu \in \mathcal{C}(\mathcal{M}_s)} \|\mu^* - \mu\|_1$ the realisability gap. Then, the performance gap is given by,*

$$||V^*_{\mu^*} - V^{\hat{\pi}}_{\mu^*}||_\infty \leq \frac{3(\epsilon_{\text{Estim}} + \epsilon_{\text{Realise}})}{(1-\gamma)^2}. \tag{5.3}$$

This theorem shows a connection between the total model deviation and the realisability gap. If the true MDP is similar to the source MDPs then we can expect a not too large of a performance loss using this framework. We can further bound the model estimation error ϵ_{Estim} given that we are in the II setting, where the novel task is part of the convex set of source tasks,

Remark 1 (Bound on L_1 Norm Difference in the Realisable Setting)**:** *It is known [60]–[62] that in the realisable setting, it is possible to bound the model estimation error term ϵ_{Estim} via the following argument. Let μ^* be the true underlying MDP, and $\hat{\mu}$ be an MLE estimate of μ^*, as defined in Theorem 7. If $\mathcal{R}$ is a bounded reward function, i.e. $r(s,a) \in [0,1], \forall (s,a)$, and ϵ_{Estim} is upper bound on the L_1 norm between $\mathcal{T}^*$ and $\hat{\mathcal{T}}$. If $n^{s,a}$ be the number of times (s,a) occur together, then with probability $1 - SA\delta$,*

$$||\mathcal{T}^* - \hat{\mathcal{T}}||_1 \leq \epsilon_{\text{Estim}} \leq \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \sqrt{\frac{2 \log\left((2^S - 2)/\delta\right)}{n^{s,a}}}.$$

Algorithm 6 Maximum Likelihood Estimation for Model-based Transfer Reinforcement Learning (MLEMTRL)

Input: weights $\boldsymbol{w}^0$, m source MDPs $\mathcal{M}_s$, data D_0, discount factor γ, iterations T.
for $t = 0, \ldots, T$ **do**
  // Stage 1: Model Estimation //
  $\boldsymbol{w}^{t+1} \leftarrow \text{Optimiser}(\log \mathbb{P}(D_t \mid \Sigma_{i=1}^{m} w_i \mu_i), \boldsymbol{w}^t)$
  Estimate the MDP: $\mu^{t+1} = \sum_{i=1}^{m} w_i \mu_i$
  // Stage 2: Model-based Planning //
  Compute the policy: $\pi^{t+1} \in \arg\max_{\pi} V^{\pi}_{\mu^{t+1}}$
  // Control //
  Observe $s_{t+1}, r_{t+1} \sim \mu^*(s_t, a_t), a_t \sim \pi^{t+1}(s_t)$
  Update the dataset $D_{t+1} = D_t \cup \{s_t, a_t, s_{t+1}, r_{t+1}\}$
end for
return An estimated MDP model μ^T and a policy π^T

From this, it can be said that the total L_1 norm then scales on the order of $\mathcal{O}(SA\sqrt{S + \log(1/\delta)}/\sqrt{T})$.

This result is specific to tabular MDPs. In tabular MDPs, the maximum likelihood estimate coincides with the empirical mean model. This result shows that in some cases, the model estimation error will shrink as more experience is collected. Finally, a trivial remark can be noted for the realisable setting (setting II),

Remark 2 (Performance Gap in the Realisable Setting)**:** *A trivial worst-case bound for the realisable case (setting II) can be obtained by setting* $\epsilon_{\text{Realise}} = 0$ *because by definition of the realisable case* $\mu^* \in \mathcal{C}(\mathcal{M}_s)$.

The theorem and the two remarks together show us a story of how the total model deviation depends on which setting we are studying through the realisability gap and how the gap may shrink with access to more data in the II setting.

The overall procedure using this framework is available in Algorithm 6.

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[2] D. Silver, A. Huang, C. J. Maddison, *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[3] D. Silver, J. Schrittwieser, K. Simonyan, *et al.*, "Mastering the game of go without human knowledge," *nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[4] B. R. Kiran, I. Sobh, V. Talpaert, *et al.*, "Deep reinforcement learning for autonomous driving: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 4909–4926, 2021.

[5] E. Leurent, "Safe and efficient reinforcement learning for behavioural planning in autonomous driving," Ph.D. dissertation, Université de Lille, 2020.

[6] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[7] S. Depeweg, J.-M. Hernandez-Lobato, F. Doshi-Velez, and S. Udluft, "Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning," in *International Conference on Machine Learning*, 2018, pp. 1192–1201.

[8] T. Pearce, M. Zaki, A. Brintrup, N. Anastassacos, and A. Neely, "Uncertainty in neural networks: Bayesian ensembling," *stat*, vol. 1050, p. 12, 2018.

[9] H. Eriksson and C. Dimitrakakis, "Epistemic risk-sensitive reinforcement learning," in *ESANN*, 2020.

[10] H. Eriksson, D. Basu, M. Alibeigi, and C. Dimitrakakis, "Sentinel: Taming uncertainty with ensemble based distributional reinforcement learning," in *Uncertainty in Artificial Intelligence*, PMLR, 2022, pp. 631–640.

[11] O. Mihatsch and R. Neuneier, "Risk-sensitive reinforcement learning," *Machine learning*, vol. 49, no. 2-3, pp. 267–290, 2002.

[12] E. Jorge, H. Eriksson, C. Dimitrakakis, D. Basu, and D. Grover, "Inferential induction: A novel framework for bayesian reinforcement learning," 2020.

[13] T. K. Buening, C. Dimitrakakis, H. Eriksson, D. Grover, and E. Jorge, "Minimax-bayes reinforcement learning," *arXiv preprint arXiv:2302.10831*, 2023.

[14] H. Eriksson, D. Basu, M. Alibeigi, and C. Dimitrakakis, "Risk-sensitive bayesian games for multi-agent reinforcement learning under policy uncertainty," *arXiv preprint arXiv:2203.10045*, 2022.

[15] H. Eriksson, D. Basu, T. Tram, M. Alibeigi, and C. Dimitrakakis, "Reinforcement learning in the wild with maximum likelihood-based model transfer," *arXiv preprint arXiv:2302.09273*, 2023.

[16] D. P. Bertsekas, *Nonlinear programming*. Athena scientific Belmont, 1999.

[17] C. Dimitrakakis and R. Ortner, *Decision making under uncertainty and reinforcement learning*, 2018.

[18] W. Thompson, "On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of two Samples," *Biometrika*, vol. 25, no. 3-4, pp. 285–294, 1933.

[19] M. Strens, "A bayesian framework for reinforcement learning," in *ICML*, 2000, pp. 943–950.

[20] I. Osband, D. Russo, and B. Van Roy, "(more) efficient reinforcement learning via posterior sampling," in *Advances in Neural Information Processing Systems*, 2013, pp. 3003–3011.

[21] I. Osband and B. Van Roy, "Model-based reinforcement learning and the eluder dimension," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[22] I. Osband and B. Van Roy, "Why is posterior sampling better than optimism for reinforcement learning?" In *International conference on machine learning*, PMLR, 2017, pp. 2701–2710.

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

[24] R. Dearden, N. Friedman, and D. Andre, "Model based Bayesian exploration," in *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, 1999, pp. 150–159.

[25] I. Osband, B. Van Roy, and Z. Wen, "Generalization and exploration via randomized value functions," in *International Conference on Machine Learning*, PMLR, 2016, pp. 2377–2386.

[26] B. O'Donoghue, I. Osband, R. Munos, and V. Mnih, "The uncertainty bellman equation and exploration," in *International Conference on Machine Learning*, 2018, pp. 3836–3845.

[27] M. Fellows, K. Hartikainen, and S. Whiteson, "Bayesian bellman operators," *Advances in Neural Information Processing Systems*, vol. 34, pp. 13 641–13 656, 2021.

[28] T. Minka, "Inferring a gaussian distribution," *Media Lab Note*, 1998.

[29] T. Minka, "Bayesian linear regression," Citeseer, Tech. Rep., 2000.

[30] M. G. Bellemare, W. Dabney, and R. Munos, "A distributional perspective on reinforcement learning," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, JMLR. org, 2017, pp. 449–458.

[31] M. Hessel, J. Modayil, H. Van Hasselt, *et al.*, "Rainbow: Combining improvements in deep reinforcement learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018.

[32] Y. Tang and S. Agrawal, "Exploration by distributional reinforcement learning," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 2710–2716.

[33] R. Dearden, N. Friedman, and S. Russell, "Bayesian q-learning," *Aaai/iaai*, vol. 1998, pp. 761–768, 1998.

[34] T. Morimura, M. Sugiyama, H. Kashima, H. Hachiya, and T. Tanaka, "Nonparametric return distribution approximation for reinforcement learning," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 799–806.

[35] W. R. Clements, B.-M. Robaglia, B. Van Delft, R. B. Slaoui, and S. Toth, "Estimating risk and uncertainty in deep reinforcement learning," *arXiv preprint arXiv:1905.09638*, 2019.

[36] H. Eriksson and C. Dimitrakakis, "Epistemic risk-sensitive reinforcement learning," *arXiv preprint arXiv:1906.06273*, 2019.

[37] Y. Chow, A. Tamar, S. Mannor, and M. Pavone, "Risk-sensitive and robust decision-making: A cvar optimization approach," in *Advances in Neural Information Processing Systems*, 2015, pp. 1522–1530.

[38] A. Tamar, Y. Glassner, and S. Mannor, "Optimizing the cvar via sampling," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[39] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath, "Coherent measures of risk," *Mathematical finance*, vol. 9, no. 3, pp. 203–228, 1999.

[40] I. Osband, C. Blundell, A. Pritzel, and B. Van Roy, "Deep exploration via bootstrapped dqn," in *Advances in neural information processing systems*, 2016, pp. 4026–4034.

[41] B. O'Donoghue, "Efficient exploration via epistemic-risk-seeking policy optimization," *arXiv preprint arXiv:2302.09339*, 2023.

[42] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *2018 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2018, pp. 3803–3810.

[43] A. A. Rusu, S. G. Colmenarejo, C. Gulcehre, *et al.*, "Policy distillation," *arXiv preprint arXiv:1511.06295*, 2015.

[44] A. Zhang, H. Satija, and J. Pineau, "Decoupling dynamics and reward for transfer learning," *arXiv preprint arXiv:1804.10689*, 2018.

[45] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine, "Reinforcement learning with deep energy-based policies," in *International conference on machine learning*, PMLR, 2017, pp. 1352–1361.

[46] E. Wei, D. Wicke, D. Freelan, and S. Luke, "Multiagent soft q-learning," *arXiv preprint arXiv:1804.09817*, 2018.

[47] C. Dimitrakakis, "Robust bayesian reinforcement learning through tight lower bounds," in *European Workshop on Reinforcement Learning (EWRL 2011)*, 2011, pp. 177–188.

[48] A. Tamar, D. Di Castro, and S. Mannor, "Policy gradients with variance related risk criteria," in *Proceedings of the twenty-ninth international conference on machine learning*, 2012, pp. 387–396.

[49] Y. Chow and M. Ghavamzadeh, "Algorithms for cvar optimization in mdps," in *Advances in neural information processing systems*, 2014, pp. 3509–3517.

[50] C.-J. Hoel, K. Wolff, and L. Laine, "Ensemble quantile networks: Uncertainty-aware reinforcement learning with applications in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[51] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath, "Coherent measures of risk," *Mathematical Finance*, vol. 9, no. 3, pp. 203–228, 1999.

[52] S. Mannor, O. Mebel, and H. Xu, "Lightning does not strike twice: Robust mdps with coupled uncertainty," *arXiv preprint arXiv:1206.4643*, 2012.

[53] W. Wiesemann, D. Kuhn, and B. Rustem, "Robust markov decision processes," *Mathematics of Operations Research*, vol. 38, no. 1, pp. 153–183, 2013.

[54] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in neural information processing systems*, 2000, pp. 1057–1063.

[55] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[56] T.-H. Ho, C. Camerer, and K. Weigelt, "Iterated dominance and iterated best response in experimental" p-beauty contests"," *The American Economic Review*, vol. 88, no. 4, pp. 947–969, 1998.

[57] G. W. Brown, "Iterative solution of games by fictitious play," *Activity analysis of production and allocation*, vol. 13, no. 1, pp. 374–376, 1951.

[58] P. Langley, "Transfer of knowledge in cognitive systems," in *Talk, workshop on Structural Knowledge Transfer for Machine Learning at the Twenty-Third International Conference on Machine Learning*, 2006.

[59] A. Lazaric, "Transfer in reinforcement learning: A framework and a survey," *Reinforcement Learning: State-of-the-Art*, pp. 143–173, 2012.

[60] A. L. Strehl and M. L. Littman, "A theoretical analysis of model-based interval estimation," in *Proceedings of the 22nd international conference on Machine learning*, 2005, pp. 856–863.

[61] P. Auer, T. Jaksch, and R. Ortner, "Near-optimal regret bounds for reinforcement learning," *Advances in neural information processing systems*, vol. 21, 2008.

[62] J. Qian, R. Fruit, M. Pirotta, and A. Lazaric, "Concentration inequalities for multinoulli random variables," *arXiv preprint arXiv:2001.11595*, 2020.