# ABSTRACT

Title of Dissertation: Robust Reinforcement Learning via Risk-Sensitivity

Erfaun Noorani
Doctor of Philosophy, 2023

Dissertation Directed by: Professor John S. Baras
Department of Electrical and Computer Engineering

The objective of this research is to develop robust-resilient-adaptive Reinforcement Learning (RL) systems that are generic, provide performance guarantees, and can generalize-reason-improve in complex and unknown task environments. To achieve this objective, we focus on exploring the concept of Risk-sensitivity in RL systems and its extensions to Multi-Agent (MA) systems. The development of robust reinforcement learning algorithms is crucial to address challenges such as model misspecification, parameter uncertainty, disturbances, and more. Risk-sensitive methods offer an approach to developing robust RL algorithms by hedging against undesirable outcomes in a probabilistic manner. The robustness properties of risk-sensitive controllers have long been established. We investigate risk-sensitive RL (as a generalization of risk-sensitive stochastic control), by theoretically analyzing the risk-sensitive exponential (exponential of the total reward) criteria and the benefits and improvements the introduction of risk-sensitivity brings to conventional RL.

By considering exponential criteria as risk measures, we aim to enhance the reliability

of our decision-making process. We explore the exponential criteria to better understand its representation, the implications of its optimization, and the behavioral characteristics exhibited by an agent optimizing this criterion. We demonstrate the advantages of utilizing exponential criteria for the development of RL algorithms. We then shift our focus to developing algorithms that effectively leverage these exponential criteria. To do that, we first focus on developing risk-sensitive RL algorithms within the framework of Markov Decision Processes (MDPs). We then broaden our scope by exploring the application of the Probabilistic Graphical Models (PGM) framework for developing risk-sensitive algorithms. Within this context, we delve into the PGM framework and examine its connection with the MDP framework. We proceed by exploring the effects of risk sensitivity on trust, collaboration, and cooperation in multi-agent systems. To conclude, we finally investigate the concept of risk sensitivity and the robust properties of risk-sensitive algorithms in decision-making and optimization domains beyond RL. Specifically, we focus on safe RL using risk-sensitive filters. Through our exploration, we seek to enhance the understanding and applicability of risk-sensitive approaches in various domains.

# Robust Reinforcement Learning
# via
# Risk-Sensitivity

by

Erfaun Noorani

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2023

Advisory Committee:
Professor John S. Baras, Chair/Advisor
Professor Eyad H. Abed
Professor Kaiqing Zhang
Professor Sanghamitra Dutta
Professor Michael C. Fu, Dean's Representative

# Dedication

To my family, my unwavering support;

To my advisor, my guiding light;

To my friends, my constant motivation;

To my colleagues and mentors, for fostering a stimulating environment;

With heartfelt gratitude, this thesis is dedicated to all of you.

# Acknowledgments

I am deeply grateful to all the individuals who have contributed to the realization of this thesis, making my graduate experience truly remarkable and unforgettable.

First and foremost, I express my heartfelt appreciation to my advisor, John Baras, who entrusted me with invaluable opportunities to engage in challenging and fascinating projects. His support and guidance have been instrumental throughout my journey. Working alongside such an exceptional mentor has been an absolute pleasure, and I am grateful for the wealth of knowledge and skills I have acquired under his tutelage.

I extend my sincere thanks to Professor Eyad Abed, Professor Michael Fu, Professor Sanghamitra Dutta, and Professor Kaiqing Zhang for graciously agreeing to serve on my thesis committee and for devoting their time and expertise to this endeavor.

I would like to take the opportunity to acknowledge the support of the funding agencies. My dissertation research has been partially supported by the Office of Naval Research (ONR) grant N000141712622, "Intelligent and Learning Autonomous Systems: Composability and Correctness", by the Army Research Lab, grants W911NF-17-2-0181, "Autonomous Resilient Cognitive Heterogeneous Swarms" (ARCHES), and W911NF-23-2-0040, "Data-Driven Engineering Research" (DataDrivER), and by the Clark Foundation. I would like to express my gratitude to the Palo Alto Research Center (PARC-A Xerox Company) and the Army Research Lab (ARL) for providing me with valuable internship and visiting opportunities. I would also like to extend

my sincere gratitude to Karl H. Johansson and the Division of Decision and Control Systems at KTH Royal Institute of Technology for their kind hospitality during my visit.

I would like to express my sincere thanks and appreciation to all my collaborators for their invaluable contributions and support. My fellow colleagues and friends have greatly enriched my graduate life, and their presence deserves special recognition for the joyous camaraderie and shared experiences we have had. Their support and guidance have significantly contributed to my personal and professional growth.

I also wish to acknowledge the valuable assistance and support provided by Mrs. Kim Edwards and the department's staff for helping me with administrative matters.

Lastly, I am indebted to my family - my mother, father, and siblings, whose unwavering support and guidance have been my anchor throughout my academic journey. They have stood by me through thick and thin, providing the strength to overcome even the most daunting challenges. Words cannot adequately express the depth of gratitude I feel towards them.

Table of Contents

# List of Figures

# Chapter 1: Introduction

Robustness is a fundamental aspect of trustworthy AI systems, as highlighted by the European Commission's Ethics Guidelines for Trustworthy AI. These guidelines emphasize the equal importance of seven requirements that support each other and must be implemented and evaluated throughout the lifecycle of an AI system. Trustworthy AI aims to maximize the benefits of AI while mitigating the associated risks.

The significance of robustness is particularly evident in today's complex systems, especially in autonomous technologies. Insufficient robustness in the design of these systems leads to accidents, failures, financial disruptions, and ethical concerns. Therefore, it is crucial to enhance the robustness of algorithms and systems to ensure their safe and reliable operation in real-world scenarios.

Several incidents illustrate the consequences of non-robust AI systems. Drone flyaway incidents, accidents in Amazon warehouses, crashes involving Tesla's autopilot and Uber's self-driving cars, the 737 MAX crashes, the "Flash Crash" of 2010, and controversies surrounding chatbots all exemplify the failures resulting from inadequate robustness.

Reinforcement learning (RL) algorithms are particularly susceptible to fragility and lack of robustness. For instance, AI systems like AlphaGo demonstrate varying performance on different board sizes, indicating their brittleness. Robotics applications, such as object grasping, reveal

that a robot proficient at handling regular objects may struggle or fail when faced with irregular or slippery objects due to the lack of robustness in their learned grasping policies. Even in simpler systems, such as an inverted pendulum, the non-robust nature of RL algorithms becomes apparent, as evidenced by the cart pole example.

Researchers have proposed various approaches to address these shortcomings. These approaches include risk-sensitive and robust RL algorithms, as well as various regularized objectives. Notably, there exists a complex connection between risk-sensitive, distributionally robust, and regularized objectives. Here we leverage risk sensitivity to enhance the overall performance of RL algorithms. Humans also exhibit risk-sensitive decision-making in various scenarios.

**Illustrative Example– Unveiling the Essence of Risk-Aware Decision Making:** Micromort [1] is a unit of risk that measures the risk of day-to-day activities and is defined as a one-in-a-million chance of sudden death. A simple back-of-a-napkin calculation based on the motor vehicle fatality statistics published by the National Highway Traffic Safety Administration in the United States shows that the chance of sudden death due to motor vehicle crashes is a fraction of a Micromort per day. Yet, we all make the decision to drive our personal transportation vehicles, take public transportation, or ride with a friend without almost any hesitation, and we choose to continue our lives without too much regard for such possible (not very likely) scenarios. Despite full knowledge of the possibility of such unfortunate events, we do not even accept staying at home or taking a walk as a viable option to prevent or reduce our chances of getting involved in a fatal accident. We even have gone to the extent to consider people who do plan their lives based on such unlikely events as having irrational fear and phobia. A Robust (worst-case) solution, e.g. H-infinity and H2 control designs [2], is appealing since it provides guarantees in the presence of uncertainty, however, a worst-case objective might lead to overly conservative

policies that might render impractical for real-world applications. The risk-sensitive nature of human decision-making has been long established by Prospect Theory [3, 4].

The empirical success of Reinforcement Learning (RL) algorithms in recent years, starting with video games [5], has fueled up further research in this field. RL algorithms are data-driven approaches to synthesize optimal policies (control laws) to optimize the system performance in unknown or complex stochastic task environments. Classical (risk-neutral) RL algorithms [6] have proven brittle (i.e., non-robust) since the early days of RL [7, 8, 9]. In stochastic decision systems, where uncertainty leads to risk (variability) in a desired performance metric, solving a stochastic optimal control task (viz., reinforcement learning applications) optimizing a risk-neutral objective (a long-run average), often leads to control policies that might perform poorly, especially in real-world applications. This is due to the fact that risk-neutral objectives typically consist of a long-run the expectation of the desired metric (average performance) which have been shown to be non-robust to noise and model uncertainties [6]. This is a well-known fact that standard RL algorithms are non-robust and brittle. This phenomenon is observed in widely-used RL algorithms, such as Actor-Critic methods, which are often unable to maintain their performance under slight variations in the environment at the testing time. Figure 1.1 shows the training and testing performance of an Actor-Critic agent in an inverted pendulum problem (see Section 5.5.2) with perturbed model parameters. While training is conducted with a given pole length, the performance of the trained agent is evaluated in a set of environments with different pole lengths. It is clear that, in the risk-neutral case, the change in the pole length results in significant performance degradation.

There is a need for robust RL algorithms that can cope with model misspecification, parameter uncertainty, disturbances, etc. Robustness is a key enabler of real-world RL applications.

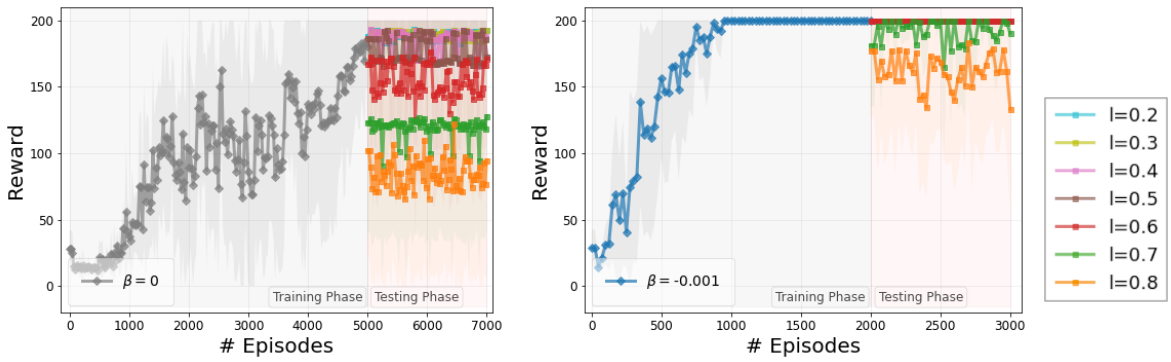



Figure 1.1: Generalization performance with respect to perturbations in the model parameters. Risk-neutral and risk-sensitive actor-critic reinforcement learning algorithms have been trained in the Cart-Pole environment with pole length $l = 0.5$, and tested for different pole length values $l \in [0.2, 0.8]$. Average reward and 90% confidence intervals over a running window of 10 episodes are depicted.

Risk-sensitive methods offer an approach to developing robust RL algorithms by hedging against undesirable outcomes in a probabilistic manner. Risk-sensitive RL investigates alternative optimization approaches, e.g., by incorporating constraints and alternative objective functions to induce robustness with respect to variations and uncertainties of the environment [10, 11, 12]. The robustness properties of risk-sensitive controllers have long been established. We investigate risk-sensitive Reinforcement Learning [13] (as a generalization of risk-sensitive stochastic control), by theoretically analyzing the risk-sensitive exponential (exponential of the total reward) criteria and the benefits and improvements the introduction of risk-sensitivity brings to conventional RL.

Robustness has been studied extensively in optimization and optimal control [14]. In reinforcement learning problems, where uncertainties in the system demand that distributional information is taken into account, robustness is associated with a stochastic optimization problem of the form:

$$\max_{\pi \in \Pi} \inf_{\rho \in \Psi} \mathbb{E}_{x \sim \pi, \zeta \sim \rho} \left[ R(x, \zeta) \right],$$

where $x \in X$ are the design parameters with distribution $\pi \in \Pi$, $\zeta \in Z$ is a random vector with distribution $\rho \in \Psi$ representing uncertain system parameters, and $R : X \times Z \to [0, \infty)$ is an objective (reward) function to be maximized. Here the system's sensitivity to maximum uncertainty (e.g., noise, disturbances) is maximized [15]. This problem is closely related to mini-max games [2, 16, 17].

Several risk-sensitive reinforcement learning approaches have been studied in recent years; from constructing constraint stochastic optimization problems [18, 19, 20] or approximately solving mini-max optimization problems [21], to investigate different statistical measures of the objective function [22, 23, 24, 25, 26, 27]. The latter approach often yields more favorable algorithmic implementations, since the computational problems associated with constraint optimization, and the convergence problems associated with the existence of multiple Nash equilibria are avoided. In particular, the algorithms in [24, 25, 28, 29] use the conditional value at risk for policy search, and the algorithms in [22, 23, 26, 30] use variance as the desired risk measure. Along the same directions, several algorithms have been developed based on postulated regularized objectives, such as Kullback-Leibler (KL) regularization [31, 32] or entropy regularization [33, 34, 35, 36, 37], leading to policy search methods, such as PPO [38], TRPO [39] and MPO [40].

Although these are ad-hoc approaches developed by experimental observations, there is a duality connection between KL- and entropy-regularized objectives and entropic risk measures [11, 27, 41, 42], associated with exponential criteria of the form:

$$\max_{\pi \in \Pi} J_\beta(\pi) := \frac{1}{\beta} \mathbb{E}_{x \sim \pi} \left[ \exp(\beta R(x)) \right].$$

In addition to this connection, exponential criteria are well-understood in the context of risk-sensitive control [17, 43, 44] and risk-sensitive MDP, and can lead to appealing algorithmic implementations [12].

Our objective is to explore the impact of optimizing this criterion on the distribution of the system performance metric (R) and its influence on agent behavior, including factors like robustness and sample efficiency. Once we determine that the desired properties can be attained through this criterion, it becomes vital to devise algorithms that can effectively tackle the inherent challenges presented by the non-linearity of the exponential operator in order to realize these desired properties.

## 1.1 Contributions and Organization of the Dissertation

The necessity of robust RL algorithms, along with the established link between risk-sensitive controllers and robust H-infinity controllers in scenarios involving known models, drives our primary question:

*"How can we use risk-sensitivity to provide RL algorithms with theoretical guarantees on the robustness in cases the models are not known?"*

In this exploration, we primarily focus on using exponential criteria as a risk measure. We discuss our contributions in detail here:

- We thoroughly investigate the representation, optimization implications, and behavioral characteristics of an agent that optimizes this criterion. (Part I)
  - We clarify the meaning of optimizing exponential criteria by examining it from two perspectives: large deviation theory (leading to an asymptotic interpretation) and the

theory of dual representation of risk measures (leading to a game-theoretic interpretation). (Chapter 3)

  - By doing so, we shed light on the benefits of employing exponential criteria in the development of Reinforcement Learning (RL) algorithms. These benefits include providing probabilistic guarantees and demonstrating the robustness of algorithms optimizing the exponential criteria. (Chapter 3)
  - We further establish the relationship between robust, risk-sensitive, and regularized RL algorithms. (Chapter 4)

- Our concentration then shifts towards crafting algorithms that effectively utilize exponential criteria. (Part II)
  - Initially, we focus on developing risk-sensitive RL algorithms within the framework of Markov Decision Processes (MDPs) and develop both Monte Carlo and actor-critic risk-sensitive RL algorithms using exponential criteria. (Chapter 5)
  - Expanding our scope, we explore the application of the Probabilistic Graphical Models (PGM) framework for creating risk-sensitive algorithms. In this context, we delve into the PGM framework and examine its connection with the MDP framework. (Chapter 6)
  - Additionally, we investigate the effects of risk-sensitivity on trust, collaboration, and cooperation in multi-agent systems. (Chapter 7)

- We explore the concept of risk-sensitivity and the robust properties of risk-sensitive algorithms in decision-making and optimization domains beyond Reinforcement Learning

(RL). Through our exploration, our aim is to enhance the understanding and applicability of risk-sensitive approaches in various domains. Specifically, we focus on Safe Reinforcement Learning utilizing risk-sensitive filters (Chapter 8).

**Organizations.** Chapter 1 serves as an introduction to the fundamental problem addressed in this thesis. Chapter 2 offers some contextual details and background materials, and establishes the notation used throughout the thesis. In Part I (Chapters 3 and 4), the focus is on investigating the characteristics of risk-sensitive RL algorithms and their relationship with robustness and regularization. Chapter 3 provides insights into exponential criteria as a risk measure and establishes the robust properties of risk-sensitive RL algorithms, while Chapter 4 establishes the connection between risk-sensitive exponential criteria and distributionally robust and regularized RL algorithms. Moving on to Part II (Chapters 5, 6, and 7), the MDP (Markov Decision Process) and PGM (Probabilistic Graphical Models) frameworks are employed to develop risk-sensitive algorithms. Chapter 5 develops risk-sensitive Monte Carlo and Actor-Critic algorithms utilizing exponential criteria within an MDP framework. Chapter 6 explores the development of risk-sensitive algorithms within the PGM framework. Chapter 7 explores the concept of risk-sensitivity in multi-agent systems. In Part III (Chapter 8), the concept of achieving robustness through risk sensitivity is further explored. We focus on incorporating risk sensitivity into safety filters for the safe control of autonomous systems. Finally, we conclude in Chapter 9. Overall, this thesis aims to comprehensively examine risk-sensitive controllers and their relationship with robustness.

# Chapter 2: Preliminaries: RL Using MDP

Conventionally, RL problem is modeled as an MDP [45], where the reward signal, defining the task at hand, is an extrinsic signal. MDP framework provides a powerful framework for modeling uncertainty. Here, we offer some contextual details, background materials, and establish the notation used throughout the thesis.

## 2.1 Modeling RL using MDP

An MDP (Markov Decision Process) is defined as the tuple

$$M = (\mathcal{S}, \mathcal{A}, p(s_1), P, r, \gamma)$$

, where:

- $\mathcal{S}$ represents the state space.
- $\mathcal{A}$ represents the action space.
- $p(s_1)$ is the initial state distribution, specifying the probability of starting in a state $s_1 \in \mathcal{S}$.
- $P : \mathcal{S} \times \mathcal{A} \rightarrow \Delta(\mathcal{S})$ is the transition kernel. It defines the probability distribution over successor states $s_{t+1} \in \mathcal{S}$ when an action $a_t \in \mathcal{A}$ is taken in state $s_t \in \mathcal{S}$. Here, $\Delta(\mathcal{S})$

denotes the space of probability distributions on $\mathcal{S}$.

- $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the cost function. It assigns a real-valued cost to each state-action pair, defining the task at hand.

- $\gamma \in (0, 1]$ is a discount factor, used to weigh future rewards against immediate rewards. It encodes the agent's time preferences.

The behavior of the agent in the environment is characterized by its policy. A (randomized) policy $\pi(a|s):\mathcal{S}\times\mathcal{A}\to\Delta(\mathcal{A})$ is a probability distribution over action space given a state, which prescribes the probability of taking an action $a \in \mathcal{A}$ when in state $s \in \mathcal{S}$; $\Delta(\mathcal{A})$ is the space of probability distributions on the action space $\mathcal{A}$.

At each time-step, the agent perceives the state environment $s_t \in \mathcal{S}$ and takes an action $a_t \in \mathcal{A}$ according to a policy $\pi(\cdot|s_t)$. After taking action $a_t$, the system undergoes a transition to a subsequent state $s_{t+1}$, following the transition probability $p(s_{t+1}|s_t, a_t)$. The agent then receives a reward $r_t = r(s_t, a_t)$.

The trajectory of a system up to time $t$ is represented by $\tau_t$, which is a sequence of states and actions taken by the system from the beginning until time $t$. It is denoted by

$$\tau_t = (s_1, a_1, s_2, a_2, \ldots, a_{t-1}, s_t).$$

To simplify notation, we use $\tau$ to refer to the system's trajectory over an entire episode, that is $\tau := \tau_T$, where $T$ is the length of the episode or the time horizon. The episode length $T$ is a random variable that determines the time step at which the system reaches a terminal state or satisfies a stopping condition.

The agents' policy and the system transition probabilities induce a trajectory distribution, a probability distribution over the sequence of states and actions, i.e. space of possible system's trajectories $\mathcal{T}$. The probability distribution induced over the space of the system's trajectories by following the policy $\pi$ in an environment with transition kernel $P$ is denoted by $\rho_{\pi,P}(\tau)$ and is given by

$$\rho_{\pi,P}(\tau) := p(s_1) \prod_{t=1}^{T} p(s_{t+1}|s_t, a_t)\pi(a_t|s_t) \tag{2.1}$$

The primary objective is to achieve a desired behavior, which is determined by an output variable $R$. In the context of RL, $R$ is typically defined as the discounted cumulative reward. This definition is given by the equation:

$$R := \sum_t \gamma^t r_t,$$

where $\gamma$ represents the discount factor, and $r_t$ denotes the reward obtained at time step $t$. It is important to note that the output of the system is a random variable due to the inherent stochastic nature of the system itself.

To enable a thorough analysis, it is crucial to incorporate measure-theoretic concepts when formulating the reinforcement learning (RL) problem. We first introduce these concepts before discussing the performance metric $R$ and the achievement of desired behaviors through the optimization of risk measures. By employing measure-theoretic tools, we can gain deeper insights into the implications and effects of different risk measures on the behavior exhibited by agents. This rigorous approach enables a comprehensive examination of how various risk measures influence the overall performance and outcomes within the RL framework, leading to a more insightful

understanding of the system.

### 2.1.1 RL and Measure Theory

Recall that $\mathcal{T}$ is the space of all possible trajectories (scenarios), which in general, maybe a finite set or an infinite set depending on the state and action spaces. The assumption of finite state and action spaces leads to a finite space of trajectories. Let $\mathcal{F}$ be a $\sigma$-algebra on the space of all possible trajectories $\mathcal{T}$. Also recall that the total reward over a trajectory is a mapping from the space of all trajectories to the reals $R : \mathcal{T} \to \mathbb{R}$ where $R_T(\tau)$ is the total reward over the systems' trajectory when the trajectory $\tau \in \mathcal{T}$ is realized—that is, $R$ is a random variable. Let $\mathcal{R}$ be a given linear space (vector space) of such random variables $R : \mathcal{T} \to \mathbb{R}$ which contains the constants, e.g., $L^p(\mathcal{T}, \mathcal{F}, \rho_{\pi,P})$ where $1 \leq p \leq \infty$. Of particular interest is the space of all bounded random variables, i.e., $\mathcal{R}=L^\infty(\mathcal{T}, \mathcal{F}, \rho_{\pi,P})$.

Each realization of system's trajectory $\tau$ has a probability of occurrence and an impact. The impact is the cost associated with the trajectory $R_T(\tau)$ which defines the task at hand and the agent has no influence on it. However, the probability distribution over the space of trajectories $\rho_{\pi,P}$ (cf. Eq. (2.1)) can be manipulated by the agent via its policy. The distribution is induced over the space of trajectories by the environment transition probabilities, which the agent also has no influence on, and the agent's policy. A risk measure provides a partial ranking over the space of policies based on the impact of the trajectories and the probability distribution that the policy induces over the space of trajectories.

## 2.2 Risk-Measures

The performance metric $R$, as a random variable, inherently possesses a certain degree of uncertainty and variability in its values. Consequently, attempting to optimize this random variable by precisely matching its distribution to a target distribution may often be an unrealistic and overly demanding task. To address this challenge, optimization techniques often rely on employing scalar-valued functions of the distribution, such as the expectation, to guide the optimization process. The expectation provides a summary statistic that captures the central tendency of the distribution, enabling a more tractable approach to optimization. This is employed in standard (risk-neutral) RL algorithms, where the focus is on maximizing the expected value of the performance metric.

However, in a lot of scenarios, optimizing solely based on the expectation may not fully capture the desired objectives or adequately handle extreme outcomes. While optimization of the performance metric $R$ as a random variable may be challenging, standard RL methods leverage the expectation to guide optimization. risk-sensitive RL, on the other hand, incorporates other metrics, often focusing on the tails of the distribution, to achieve more robust optimization outcomes.

### 2.2.1 Standard RL

In classical RL, the objective typically is to optimize a risk-neutral objective on a class of policies $\Pi$, that is, expectation of some long-run average, such as expected discounted cumulative

reward, i.e.,

$$J(\pi) := \mathbb{E}_{\tau \sim \rho_{\pi,P}} \Big[ R(\tau) \Big] \tag{2.2}$$

where $R(\tau) := \sum_{t=1}^{T} \gamma^t r(s_t, a_t)$ is the total discounted reward and $\mathbb{E}_{\tau \sim \rho_{\pi,P}}[\cdot]$ denotes the expectation taken over the system's trajectory generated by following the policy $\pi$ in an environment with transition kernel $P$, i.e., $s_1 \sim p_1$, $a_t \sim \pi(\cdot|s_t)$, and $s_{t+1} \sim p(\cdot|s_t, a_t)$.

### 2.2.2 Risk-Sensitive RL

Systematizing the exploration and development of risk-aware algorithmic decision-making is a necessity. In stochastic systems, uncertainty leads to stochasticity in the system's performance. The risk-aware decision algorithms evaluate actions according to a risk-measure beyond merely the expectation of the desired system performance, resulting in algorithms that seek to shape the distribution of the system performance into a desired shape beyond simply moving the expectation to a more desired value. A variety of risk-measures, e.g. the mean-variance measure [23, 26], Conditional-Value-at-Risk (CVaR) and Value-at-Risk (VaR) [24, 25], Entropic and exponential risk-measures [12, 27, 42, 46, 47, 48], to name a few, have been investigated over the years from the perspectives of a diverse set of disciplines, from Finance and Economics to Control Theory and Reinforcement Learning to Mathematical Psychology and Neuroscience. The question of the suitability of a particular risk-measure for a given task has been the subject of much debate. The axiomatic approach of the theory of Coherent risk-measures, and its relaxation to Convex risk-measures, [49, 50] are attempts to solve the much-needed systematization and categorization of the risk-measures, providing some hints on the suitability of a risk-measure.

Though, this axiomatic approach has been successful in facilitating the further development of risk-aware decision algorithms, the questions that "if" and "to-what-extend" the axioms in the theory of Coherent (and Convex) risk-measures are suitable for control tasks, e.g. robotics, have not been discussed. We first introduce the classes of coherent and convex risk measures. Subsequently, our focus primarily shifts to the exponential criterion, which serves as a specific example of convex risk-measures, for the remainder of the thesis.

#### 2.2.2.1 Coherent and Convex Risk Measures

Recall that $(\mathcal{T}, \mathcal{F}, \rho_{\pi,P})$ is a probability triple and $\mathcal{R}$ be the space of random variables defined on $\mathcal{T}$. A risk measure is a mapping from the space of random variables to the reals, i.e., $\mathcal{R} \to \mathbb{R}$, e.g. $J_\beta$ when the policy $\pi$ is given. A risk measure is said to be Convex [49] if and only if the axioms of (1) monotonicity, (2) translation invariance, and (3) convexity hold which are defined as follows, where we highlight the dependence of the objective function $J_\beta$ on the random variable $R$ by explicitly using $R$ as an argument to the function, i.e., $J_\beta(\pi, R)$ (cf. Eq. (3.2)).

For all $R$ and $R' \in \mathcal{R}$:

1. **Monotonicity.** For $R \leq R'$,

$$J_\beta(\pi, R) \leq J_\beta(\pi, R')$$

2. **Translation Invariance.** For $m \in \mathbb{R}$,

$$J_\beta(\pi, R + m) = J_\beta(\pi, R) + m$$

3. **Convexity.** For $0 \leq \alpha \leq 1$,

$$J_\beta(\pi, \alpha CR(1-\alpha)R') \leq \alpha J_{\pi,\beta}(\pi, R) + (1-\alpha) J_\beta(\pi, R')$$

A convex risk measure is called coherent if and only if in addition to the properties (1), (2), and (3), is positive homogeneous, i.e.,

4. **Positive Homogeneity.** For $\alpha \geq 0$,

$$J_\beta(\pi, \alpha R) = \alpha J_\beta(\pi, R)$$

Convex and coherent risk measures are indeed appealing due to their robust dual representation [49]. This property allows for a convenient and flexible characterization of risk measures. The dual representation of a coherent and convex risk measure provides an alternative way to express the risk measure using a mathematical formulation involving the expectations of certain functions of the random variable. This dual formulation allows for efficient computation and optimization of risk measures.

**Proposition 1** (Dual Representation of Coherent Risk Measures)**.** *A functional* $J : R \to \mathbb{R}$ *is a coherent risk measure if and only if there exists a subset* $Q$ *of* $M_{1,f}$ *such that*

$$J = \sup_{Q \in Q} E_Q[R]$$

*Moreover, Q can be chosen as a convex set for which the supremum is attained.*

**Proposition 2** (Dual Representation of Convex Risk Measures)**.** *Any convex risk measure* $J$ :

$R \to \mathbb{R}$ *is of the form*

$$J = \sup_{Q \in \mathcal{Q}} E_Q[R] - D(Q)$$

*where the penalty function D is given by*

$$D(Q) := \sup E_Q[R]$$

*Moreover,* $D$ *is the minimal penalty function which represents* $J$.

It can be verified that the risk-sensitive exponential criteria are a convex but not coherent risk measure. The remaining portion of the thesis primarily centers around the exponential criteria.

Part I

# Vital Role of Risk in Real-World Reinforcement Learning

We explore the concept of risk-sensitivity as a means of introducing robustness to our algorithmic decision-making. By considering risk measures, we aim to enhance the reliability of our decision-making process. In our exploration, we focus on exponential criteria as a risk measure. We investigate the exponential criteria to better understand its representation, the implications of its optimization, and the behavioral characteristics exhibited by an agent optimizing this criterion.

# Chapter 3: Risk-Sensitive RL Using Exponential Criteria

## 3.1 Overview

Exponential criterion is a particular example of a Convex risk measure, i.e.,

$$J_\beta(\pi) := \mathbb{E}_{\tau\sim\rho_{\pi,P}}\left[\beta e^{\beta R(\tau)}\right] \tag{3.1}$$

where $\beta \in \mathbb{R}$ is a constant design parameter. It should be noted that due to monotonicity of the logarithmic function, the optimization of the risk-sensitive objective above is equivalent to the following objective function with a risk parameter $\beta$ is also can be given by

$$J_\beta(\pi) := \frac{1}{\beta}\log \mathbb{E}_{\tau\sim\rho_{\pi,P}}\left[e^{\beta R(\tau)}\right] \tag{3.2}$$

A positive risk parameter $\beta>0$ corresponds to a risk-averse behavior and a negative risk parameter $\beta<0$ corresponds to a risk-seeking behavior. Notice that a Taylor expansion of (3.14) reveals an intuition behind how the exponential criterion incorporates risk into the objective function, since it incorporates an infinite sum of the higher moments of the return, i.e., for small $\beta$ we get:

$$\frac{1}{\beta}\log \mathbb{E}\left[e^{\beta R(\tau)}\right] = \mathbb{E}\left[R(\tau)\right] + \frac{\beta}{2}\mathrm{Var}\left[R(\tau)\right] + \mathcal{O}(\beta^2) \tag{3.3}$$

Equation (3.3) shows how the risk-sensitive parameter $\beta$ controls the weight of the moments of the cumulative reward in the objective function. Finally, we note that as the risk-sensitive parameter $\beta$ approaches zero, the exponential objective approaches the risk-neutral objective. The Taylor series representation is useful, but it alone is insufficient to clarify the advantages derived from using this criterion. Therefore, we will present some results to shed light on the interpretation of optimizing exponential criteria. These results will highlight the advantages gained in terms of robustness and reliability.

We explore the interpretation and benefits of optimizing exponential criteria from two theoretical perspectives: large deviation theory and the theory of dual representation of convex risk measures. Using large deviation theory, we demonstrate that optimizing the risk-sensitive exponential criterion is equivalent to optimizing the exponential rate of decay of the tail of the reward distribution. This approach provides an asymptotic interpretation of the optimization process. From the theory of dual representation of convex risk measures, we reveal that maximizing the exponential criterion can be formulated as a game. For a risk-averse agent, it becomes a non-cooperative (min-max) game, while for a risk-seeking agent, it transforms into a cooperative (min-min) game. This re-formulation aligns with the intuition that a risk-averse agent perceives the environment as adversarial, while a risk-seeking agent sees it as favorable.

Furthermore, we establish that risk-sensitivity is linked to the robustness of a policy. This robustness is quantified by a lower bound on the probability of achieving good performance when the transition distribution during testing deviates from the distribution induced by the trajectory distribution during training. *In summary, this exploration sheds light on the interpretation of optimizing exponential criteria and highlights the advantages of doing so in terms of robustness and reliability.*

## 3.2 Related Work

Exponential criterion [43, 51] has been the cornerstone of risk-sensitive control and risk-sensitive MDP literature and has been substantially studied over the last five decades [17, 43, 52, 53, 54, 55, 56, 57]. The exponential criterion is a mathematically convenient and intuitively appealing risk measure with a firm theoretical foundation rooted in Large Deviation Theory. Most importantly, prior work [17, 54, 55, 56, 57] has established that the exponential of an integral criterion emerges from the mathematical analysis of "four blocks" or H-infinity output robust control in its full generality. Large deviation theory is a branch of mathematics that deals with the study of rare events in random processes. For more details on large deviation theory, we refer the readers to [58, 59]. A more recent review on the subject [60] offers a more accessible treatment of the basic concepts in large deviation theory. A systematic approach to design and evaluations of risk measures for RL tasks is still an open question, but the axiomatic definition of Coherent [61] and later its relaxation Convex risk-measures [49] that provides a systematic approach to categorization and evaluation of risk-measures is an attempt to answer this question in the context of mathematical Finance and Economics literature.

## 3.3 Understanding the Optimization of Exponential Criteria

In our investigation, we delve into the interpretation of optimizing exponential criteria by examining it through two theoretical frameworks: (I) Large Deviation Theory, and (II) the theory of dual representation of convex risk measures.

### 3.3.1 Large Deviation Theory and Asymptotic Interpretation

Risk-neutral objectives fail to take into account the effect of significant trajectories with low probability which leads to brittle algorithms and non-robustness. Optimizing the risk-sensitive exponential criterion is equivalent to optimization of the exponential rate of decay of the tail of the cost distribution. That is to say, the exponential criteria take into account the tail of the distribution.

**Theorem 1.** *For a given negative risk parameter (risk-aversion)* $\beta<0$*, the maximization of the risk-sensitive exponential criterion* $J_{l_\beta}$ *in* (3.14) *is equivalent to the maximization of the exponential rate of decay of the left tail of the system's trajectory reward distribution, i.e., for a given* $\beta < 0$*, there exists a constant* $\kappa\in\mathbb{R}$ *such that*

$$\arg\max_{\pi} J_{l_\beta}(\pi) = \lim_{T\to\infty} \arg\min_{\pi} \mathbb{P}[R_T < \psi]$$

*where* $\mathbb{P}[R_T < \psi]$ *denotes the probability of the event* $R_T < \psi$*. Similarly, for a given positive risk parameter (risk-seeking)* $\beta>0$*, the maximization of the risk-sensitive exponential criterion* $J_{l_\beta}$ *in* (3.14) *is equivalent to minimization of the exponential rate of decay of the right tail of the system's trajectory reward distribution, that is, for a given* $\beta > 0$*, there exists a constant* $\psi$ *such that*

$$\arg\max_{\pi} J_{l_\beta}(\pi) = \lim_{T\to\infty} \arg\max_{\pi} \mathbb{P}[R_T > \psi]$$

*Proof.* To see that, let $P(R_t\in\Psi)$ be the probability that the total reward over the system's trajec-

tory $R_t$ takes on value in a set $\Psi$. Then $P(R_t \in \Psi)$ is said to satisfy a Large Deviation Principle (LDP) with rate $I_\Psi$ if the limit (3.4) exist.

$$\lim_{t\to\infty} \frac{1}{t} \log p(R_t \in \Psi) = -I_\Psi \tag{3.4}$$

where $I_\Psi = \inf_{\psi\in\Psi} I(\psi)$ is a positive constant and is called the rate function. Thus, in small o notation, we have

$$\frac{1}{t} \log p(R_t \in \Psi) = -I_\Psi + o(1)$$

that is to say, the dominant behaviour of $P(R_t \in \Psi)$ is a decaying exponential in $t$.

By The Gartner-Ellis theorem [59], if $\lambda_\pi(\beta) := \lim_{t\to\infty} \frac{\beta}{t} J_\beta(\pi)$ exist, then

$$I(\psi) = \sup_{\beta\in\mathbb{R}} \Big\{ \beta\psi - \lambda_\pi(\beta) \Big\}$$

where

$$\lambda_\pi(\beta) := \lim_{t\to\infty} \frac{\beta}{t} J_\beta(\pi), \quad J_\beta(\pi) := \frac{1}{\beta} \log \mathbb{E}_{\tau\sim\rho_{\pi,P}} \Big[ e^{\beta R_t} \Big]$$

Thus,

$$\frac{1}{t} \log P(R_t \in \Psi) = -\inf_{\psi\in\Psi} \sup_{\beta\in\mathbb{R}} \Big\{ \beta\psi - \lambda_\pi(\beta) \Big\} + o(1)$$

Then by noting that $\lambda_\pi$ is the Legendre-Fenchel transform of the rate function $I$, we have that for

a given $\beta>0$ (resp. $\beta<0$ ), there exist a set $\Psi=[\bar{\psi}, \infty)$ (resp. $\Psi=(-\infty, \bar{\psi}]$) such that

$$\frac{1}{t} \log p(R_t \in \Psi) = \lambda_\pi(\beta) - \beta\bar{\psi} + o(1)$$

by using small o notation, i.e.,

$$\frac{1}{t} \log p(R_t \in \Psi) = \frac{\beta}{t} J_\beta(\pi) - \beta\bar{\psi} + o(1)$$

For more details on large deviation theory, please see [60]. □

From Theorem (1), it is clear that an increasing positive value of $\beta$ corresponds to a more optimistic (risk-seeking) behavior and an increasing negative value of $\beta$ corresponds to a more pessimistic (risk-averse) behavior. The value $\beta = 0$ reduces the $J_\beta(\pi)$ to the risk-neutral expected value $J(\pi)$.

### 3.3.2 Duality and Game Theoretic Interpretation

Minimizing the exponential criterion may be expressed as a game formulation; a non-cooperative (min-max) game for the positive risk-parameter (risk-averse agent) and a cooperative (min-min) game for a negative risk-parameter (risk-seeking agent). Such re-formulation is in agreement with the intuition that a risk-averse agent is pessimistic and views the environment as a force that acts against it, while a risk-seeking agent is optimistic and views the environment as a force that acts in his favor.

It is well-known that the risk-sensitive exponential criterion (cf. Eq. (3.2)) for a positive risk parameter is a Convex risk measure and has a robust dual representation [49].

**Theorem 2.** *For a positive risk parameter $\beta<0$ (risk-aversion), the risk-sensitive exponential criterion has a dual representation given by*

$$J_\beta(\pi) = \inf_{\hat{\pi}} \left\{ \mathbb{E}_{\tau\sim\rho_{\hat{\pi},P}} \Big[ R(\tau) \Big] - \frac{T}{\beta} D_{KL}\Big( \pi_{\hat{\theta}}(\cdot|s_t), \pi_\theta(\cdot|s_t) \Big) \right\} \tag{3.5}$$

*and for a negative risk parameter $\beta>0$ (risk-seeking) the dual representation is given by*

$$J_\beta(\pi) = \sup_{\hat{\pi}} \left\{ \mathbb{E}_{\tau\sim\rho_{\hat{\pi},P}} \Big[ R(\tau) \Big] + \frac{T}{\beta} D_{KL}\Big( \pi_{\hat{\theta}}(\cdot|s_t), \pi_\theta(\cdot|s_t) \Big) \right\} \tag{3.6}$$

*where*

$$D(Q,P) = \begin{cases} \mathbb{E}_Q\Big[ \log \frac{dQ}{dP} \Big] & \text{if } Q \ll P \\ \infty & \text{otherwise} \end{cases}$$

*is the relative entropy of $Q$ w.r.t. $P$ (Kullback–Leibler (KL) divergence of probability distribution $Q$ from $P$), and the support of $\hat{\pi}$ is contained within the support of $\pi$, that is to say, $\hat{\pi}$ is absolutely continuous with respect to $\pi$.*

*Proof.* By the dual representation theorem of Convex risk measures [49, 50], the exponential criteria has a dual representation given by [49]

$$J_\beta(\theta) = \sup_{\hat{\theta}} \left\{ \mathbb{E}_{\tau\sim\rho_{\hat{\theta}}} \Big[ R(\tau) \Big] - \frac{1}{\beta} D_{KL}\Big( \rho_{\hat{\pi},P}(\tau), \rho_{\pi,P}(\tau) \Big) \right\}$$

where the support of $\rho_{\hat{\pi},P}$ is contained within the support of $\rho_{\pi,P}$, that is to say, $\rho_{\hat{\pi},P}$ is absolutely continuous with respect to $\rho_{\pi,P}$. Note that the support of a probability distribution $\rho_\theta$ is defined as the set $\{\tau \in \mathcal{T} | \rho_{\pi,P}(\tau)>0\}$.

The dual representation of the exponential criteria presented here is an application of the theorem of dual representation of Convex risk measures [49, 50] for the special case of entropic risk measure. For the general statement of the theorem, proof, and detailed explanation, please see [49, 50] and references therein.

By noting the definition of the trajectory distribution $\rho_\theta$ (cf. Eq (2.1)), it can be shown that, see Theorem 5 in [62],

$$D_{KL}\Big(\rho_{\hat{\theta}}(\tau), \rho_\theta(\tau)\Big) = T\mathbb{E}_{\tau\sim\rho_{\hat{\theta}}}\Big[D_{KL}\Big(\pi_{\hat{\theta}}(\cdot|s_t), \pi_\theta(\cdot|s_t)\Big)\Big]$$

where $T$ is the time horizon. Thus, we have

$$J_\beta(\theta) = \sup_{\hat{\theta}} \mathbb{E}_{\tau\sim\rho_{\hat{\theta}}}\Big[R(\tau) - \frac{T}{\beta} D_{KL}\Big(\pi_{\hat{\theta}}(\cdot|s_t), \pi_\theta(\cdot|s_t)\Big)\Big] \tag{3.7}$$

where the support of $\pi_{\hat{\theta}}$ is contained within the support of $\pi_\theta$. Note that such conditions can be satisfied, for example, if the policies are always non-zero for all actions given a state. □

**Remark 1.** *by taking $\pi = \hat{\pi}$ in the dual representation (cf. Eq. (3.5) and (3.6)), it is clear that for $\beta>0$ (risk-averse)*

$$J_\beta(\pi) \geq J(\pi) \quad \forall \pi \in \Pi$$

*and for $\beta<0$ (risk-seeking) agent*

$$J_\beta(\pi) \leq J(\pi) \quad \forall \pi \in \Pi$$

*Also, the risk-neutral objective is a limit case of the risk-sensitive exponential criterion as* $\beta \to 0$, *i.e.,* $\lim_{\beta \to 0} J_\beta(\pi) = J(\pi)$

From Theorem (2), we have that for a positive risk parameter $\beta > 0$ (risk-aversion), the minimization of risk-sensitive criterion is equivalent to a non-cooperative (minimax) game and for a negative risk parameter $\beta < 0$ (risk-seeking) is equivalent to a cooperative (minimin) game. To see that, take the minimum over the policy class in Eq. (3.5) and (3.6).

## 3.4 The Benefits of Optimizing the Exponential Objective: A Robustness Analysis

Risk-sensitivity in reinforcement learning is often associated with the following general problem:

$$\max_{\pi} \inf_{\rho_\theta \in \Psi} \mathbb{E}_{\tau \sim \rho_\theta} \Big[ R(\tau(\theta)) \Big], \tag{3.8}$$

which induces distributional robustness with respect to the probability distribution over the possible trajectories $\mathcal{T}$. Maximization over the parameter space $\theta \in \mathbb{R}^d$ simulates optimization over all policies $\pi \in \Pi$. Minimization over the distributions $\rho_\theta$ corresponds to reducing the sensitivity of the uncertainties that affect $\rho_\theta$, which include both the initial state distribution $p_0$, and the transition probabilities $P$, i.e., all uncertainties with respect to the model parameters and any noise perturbation of the system dynamics. Typically the space $\Psi$ is constrained to a closed set of distributions that defines a trade-off between optimality and conservativeness of the policy. However, solving (3.8) with dynamic programming and game theoretic methods becomes intractable in large state/action spaces, and methods that approximate its solution have been stud-

ied [19, 20, 21], including the use of different statistical measures of the objective function to avoid the minimization over the distributions $\rho_\theta$ [22, 23, 24, 25, 26].

We focus on the following definition of a risk-sensitive reinforcement learning problem that incorporates an inherent regularization term for the set of distributions $\rho_\theta$:

$$\max_{\theta} \inf_{\rho_\theta \in \Psi} -\mathrm{sgn}(\beta)\left\{\mathbb{E}_{\tau\sim\rho_\theta}\left[R(\tau(\theta))\right] - \frac{1}{\beta} D\left(\rho_\theta, \bar{\rho}_\theta\right)\right\}, \tag{3.9}$$

where $D(Q, P)$ represents the Kullback-Leibler divergence measure:

$$D(Q,P) = \begin{cases} \mathbb{E}_Q\left[\ln \frac{dQ}{dP}\right] & \text{if } Q \ll P \\ \infty & \text{otherwise} \end{cases}. \tag{3.10}$$

The baseline distribution $\bar{\rho}_\theta$ is an independent parameter of the optimization formulation and can incorporate prior information and constraints of the problem. The use of baseline terms in reinforcement learning is widely adopted [6, 63] and is further explained in Section 5.3. The parameter $\beta$ is the risk-sensitive parameter that controls the behavior of the agent and the weight of the regularization term. In particular, $\beta > 0$ induces a risk-seeking (optimistic) approach, while $\beta < 0$ invokes a risk-averse (pessimistic) approach. The value of $\beta$ determines the trade-off between optimizing with respect to the observed reward, or with respect to staying close to the baseline trajectory distribution $\bar{\rho}_\theta$, which is a trade-off between optimality and conservativeness of the policy.

Problem (3.9) is still a game-theoretic formulation of the risk-sensitive reinforcement learning problem, which can be hard to solve directly. However, it is well known (see, e.g., [49, 64,

65]), that the following duality relationship, with respect to a Legendre-type transform, holds:

**Theorem 1.** *Consider a measurable space* $(\Omega, \mathcal{F})$*, where* $\mathcal{F}$ *is a* $\sigma$*-algebra on* $\Omega$*. Let* $\mathcal{P}(\Omega)$ *be a set of probability measures* $P : \Omega \to [0, 1]$*, and* $P_\mu, P_\nu \in \mathcal{P}(\Omega)$*. In addition, consider a bounded measurable function* $Z : \Omega \to \mathbb{R}$*. Then the free energy is defined as:*

$$J_{l_\beta}(Z) = \frac{1}{\beta} \ln \mathbb{E}_{P_\mu}\left[e^{\beta Z}\right] \tag{3.11}$$

*and the KL divergence measure:*

$$D_{KL}(P_\nu, P_\mu) = \begin{cases} \int \ln(\frac{dP_\nu}{dP_\mu}) dP_\nu & \text{if } C_{KL}(P_\nu, P_\mu) \\ \infty & \text{otherwise} \end{cases} \tag{3.12}$$

*are in duality with respect to a Legendre-type transform, in the following sense:*

$$J_{l_\beta}(Z) = \begin{cases} \sup_{P_\nu \in \mathcal{P}(\Omega)} \left\{ \mathbb{E}_{P_\nu}[Z] - \frac{1}{\beta} D_{KL}(P_\nu, P_\mu) \right\}, & \beta > 0 \\ \inf_{P_\nu \in \mathcal{P}(\Omega)} \left\{ \mathbb{E}_{P_\nu}[Z] - \frac{1}{\beta} D_{KL}(P_\nu, P_\mu) \right\}, & \beta < 0 \end{cases} \tag{3.13}$$

*Here the conditions* $C_{KL}(P_\nu, P_\mu)$ *include* $P_\nu \ll P_\mu$ *and* $\int \ln(\frac{dP_\nu}{dP_\mu}) dP_\nu \in L^1(P_\nu)$.

As an immediate result of Theorem 1, we get the following corollary:

**Corollary 2.1.** *The problem:*

$$\max_\theta J_{l_\beta}(\theta) := \frac{1}{\beta} \ln \mathbb{E}_{\tau \sim \rho_\theta}\Big[\exp(\beta R(\theta))\Big] \tag{3.14}$$

*is equivalent to* (3.9)*, for the baseline distribution* $\bar{\rho}_\theta = \rho_\theta$*, i.e., the current trajectory distribution*

*of the algorithm, assuming that the maximum is attained.*

### 3.4.1 Policy Robustness

Given an MDP $\mathcal{M}=(\mathcal{S}, \mathcal{A}, p_0, P, r, \gamma)$ with transition probabilities $P$, a fixed policy $\pi$, parameterized by $\theta$, defines a trajectory distribution $\rho_\theta$ given by (2.1). RL algorithms try to find the optimal policy $\pi(\theta)$ given observations of the rewards $r$ of $\mathcal{M}$. However, during the testing phase when the policy $\pi(\theta)$ is applied, environment and model perturbations may alter the transition probabilities. Thus, the agent is asked to operate on a perturbed MDP $\hat{\mathcal{M}}=(\mathcal{S}, \mathcal{A}, \hat{p}_0, \hat{P}, r, \gamma)$, where $\hat{P}$ represents the perturbed system of transition probabilities. This phenomenon is especially present when training takes place in simulation environments while testing is transferred to a real system.

In this case, risk-sensitivity can be associated with a measure of robustness of a policy $\pi(\theta)$, quantified by a lower bound on the probability of good performance when the transition distribution $\hat{\rho}$ during testing deviates from the distribution $\rho_\theta$ induced by $\pi(\theta)$. In this work, we will adopt the following definition of robustness of a policy $\pi(\theta)$.

**Definition 1.** *Let $\pi(\theta)$ be a given policy and $\rho_\theta$ be its associated trajectory distribution given by (2.1) with transition probabilities $P$. In addition, let $\hat{\rho}$ be a trajectory distribution generated by $\pi(\theta)$ given a perturbed system of transition probabilities $\hat{P}$. The policy $\pi(\theta)$ is $(\xi, \delta, \epsilon)$-robust if, for $\delta, \epsilon > 0$, and under the condition $D(\hat{\rho}, \rho_\theta) \leq \epsilon$, it holds that*

$$\mathbb{P}_{\tau\sim\hat{\rho}}\left[R(\tau(\theta)) > \xi\right] \geq 1 - \delta(\xi, \epsilon), \tag{3.15}$$

*where $D(\cdot, \cdot)$ represents the KL divergence defined in (3.10).*

In general, non-trivial sets of parameters $(\xi, \epsilon, \delta)$ such that the condition (3.15) is satisfied cannot be found. However, for optimal policies with respect to problem (3.14), we can analytically provide such parameters using standard concentration inequalities. First, Theorem 2 provides upper bounds on the probability of the tails of the cumulative rewards $R$, in the case of bounded reward, i.e., $R \leq R_{max}$ almost surely. Note that $R_{max} = \frac{r_{max}(1-\gamma^T)}{1-\gamma}$ when the per step reward is bounded $r \leq r_{max}$.

**Theorem 2.** *Let $\pi(\theta^*)$ be an optimal policy with respect to (3.14), i.e., $\pi(\theta^*) = \arg\max_\theta J_{l_\beta}(\theta)$, and $\rho_{\theta^*}$ be its associated trajectory distribution given by (2.1) with transition probabilities $P$. In addition, let $\hat{\rho}$ be a trajectory distribution generated by $\pi(\theta)$ given a perturbed system of transition probabilities $\hat{P}$ such that $D(\hat{\rho}, \rho_{\theta^*}) \leq \epsilon$. Then the following inequalities hold:*

$$\mathbb{P}_{\tau\sim\hat{\rho}}[R(\tau) \leq \xi] \leq \frac{R_{max}}{R_{max}-\xi}\left(1 - \frac{1}{R_{max}}J^*_{l_\beta} + \frac{\epsilon}{|\beta|R_{max}}\right), \qquad \beta < 0, \tag{3.16}$$

$$\mathbb{P}_{\tau\sim\hat{\rho}}[R(\tau) \geq \xi] \leq \frac{1}{\xi}J^*_{l_\beta} + \frac{\epsilon}{\beta\xi}, \qquad \beta > 0, \tag{3.17}$$

*where* $J^*_{l_\beta} = \frac{1}{\beta}\ln \mathbb{E}_{\tau\sim\rho_{\theta^*}}\Big[\exp(\beta R(\tau))\Big]$.

*Proof.* For (3.17), using Markov's inequality, we get:

$$\begin{aligned}
\mathbb{P}_{\tau\sim\hat{\rho}}[R(\tau) \geq \xi] &\leq \frac{\mathbb{E}_{\tau\sim\hat{\rho}}[R(\tau)]}{\xi} \\
&\leq \frac{1}{\xi}\left(J^*_{l_\beta} + \frac{1}{\beta}D(\hat{\rho}, \rho_{\theta^*})\right) \\
&\leq \frac{1}{\xi}J^*_{l_\beta} + \frac{\epsilon}{\beta\xi}
\end{aligned} \tag{3.18}$$

where we have used the duality relationship (3.13) for $\beta > 0$ to get:

$$J^*_{l_\beta} = \frac{1}{\beta} \ln \mathbb{E}_{\tau\sim\rho_{\theta^*}} \Big[ \exp(\beta R(\tau)) \Big] = \sup_{\rho} \Big\{ \mathbb{E}_{\tau\sim\rho} [R(\tau)] - \frac{1}{\beta} D(\rho, \rho_{\theta^*}) \Big\}$$
$$\geq \mathbb{E}_{\tau\sim\hat{\rho}} [R(\tau)] - \frac{1}{\beta} D(\hat{\rho}, \rho_{\theta^*}),$$

which implies that $\mathbb{E}_{\tau\sim\hat{\rho}} [R(\tau)] \leq J^*_{l_\beta} + \frac{1}{\beta} D(\hat{\rho}, \rho_{\theta^*})$.

Similarly, for (3.16), using reverse Markov's inequality and assuming that $R < R_{max}$, a.s., we get:

$$\begin{aligned} \mathbb{P}_{\tau\sim\hat{\rho}} [R(\tau) \leq \xi] &\leq \frac{R_{max} - \mathbb{E}_{\tau\sim\hat{\rho}}[R(\tau)]}{R_{max} - \xi} \\ &\leq \frac{R_{max}}{R_{max} - \xi} \left( 1 - \frac{1}{R_{max}} J^*_{l_\beta} - \frac{1}{R_{max}\beta} D(\hat{\rho}, \rho_{\theta^*}) \right) \\ &\leq \frac{R_{max}}{R_{max} - \xi} (1 - \frac{1}{R_{max}} J^*_{l_\beta} + \frac{\epsilon}{|\beta| R_{max}}) \end{aligned} \tag{3.19}$$

where we have used the duality relationship (3.13) for $\beta < 0$ to get:

$$J^*_{l_\beta} = \frac{1}{\beta} \ln \mathbb{E}_{\tau\sim\rho_{\theta^*}} \Big[ \exp(\beta R(\tau)) \Big] = \inf_{\rho} \Big\{ \mathbb{E}_{\tau\sim\rho} [R(\tau)] - \frac{1}{\beta} D(\rho, \rho_{\theta^*}) \Big\}$$
$$\leq \mathbb{E}_{\tau\sim\hat{\rho}} [R(\tau)] - \frac{1}{\beta} D(\hat{\rho}, \rho_{\theta^*}), \ \beta < 0,$$

which implies that $-\mathbb{E}_{\tau\sim\hat{\rho}} [R(\tau)] \leq -J^*_{l_\beta} - \frac{1}{\beta} D(\hat{\rho}, \rho_{\theta^*})$. □

**Remark 2.** *Note that in Theorem 2, the term* $J^*_{l_\beta} = \frac{1}{\beta} \ln \mathbb{E}_{\tau\sim\rho_{\theta^*}} \Big[ \exp(\beta R(\tau)) \Big]$ *does not depend on the perturbed system of transition probabilities* $\hat{P}$ *in the test environment.*

Equations (3.16) and (3.17) give upper bounds on the probability of the two tails of the cumulative rewards R. In particular, a risk-averse agent tries to optimize for the maximum average reward weighing in the maximization of the decay of the left tail of the distribution of the total

reward, while a risk-seeking agent weights in the maximization of the decay of the right tail of the reward distribution. This is consistent with the following theorem proven in [44] using the Gartner-Ellis theorem of Large Deviation:

Given the results of Theorem 2, we can show that the risk-averse policy ($\beta < 0$) is a $(\xi, \delta, \epsilon)$-robust policy according to Definition 1. This is shown in Corollary 2.2.

**Corollary 2.2.** *Let an optimal policy* $\pi(\theta^*) = \arg\max_\theta J_{l_\beta}(\theta)$ *with respect to* (3.14) *for* $\beta < 0$. *Then,* $\pi(\theta^*)$ *is* $(\xi, \delta, \epsilon)$*-robust according to Definition 1 with:*

$$\delta(\xi, \epsilon) = \frac{R_{max}}{R_{max} - \xi}(1 - \frac{1}{R_{max}} J^*_{l_\beta} + \frac{\epsilon}{|\beta| R_{max}}). \tag{3.20}$$

*In addition, for a given* $\delta = \bar{\delta}$*, we can quantify* $\xi$ *by:*

$$\xi = R_{max} - \frac{R_{max}}{\bar{\delta}}(1 - J^*_\beta + \frac{\epsilon}{\beta R_{max}}). \tag{3.21}$$

*Proof.* It follows directly from (3.16) since $\mathbb{P}[R > \xi] = 1 - \mathbb{P}[R \leq \xi]$. □

So far, we have shown how the optimization problem (3.14) is connected to risk-sensitivity and robustness of the learned policy with respect to model perturbations. However, as will be discussed in Section 5.4.2, the presence of the non-linearity introduced by the logarithm in (3.14) creates computational problems in algorithmic implementations. For this reason, throughout the rest of this paper we will study the equivalent (in terms of optimal policy) problem:

$$\max_\theta J_\beta(\theta) := \frac{1}{\beta} \mathbb{E}_{\tau \sim \rho_\theta} \Big[ \exp(\beta R(\theta)) \Big]. \tag{3.22}$$

x

# Chapter 4: Exploring The Connection Between Robustness, Risk-Sensitivity, and Regularization

## 4.1 Overview

There exists an interplay between risk-sensitivity, robustness, and regularization—a relationship that warrants thorough exploration. By delving into this intricate connection, we can uncover valuable insights into their influence on various domains and scenarios. Recall that in classical RL, the objective typically is to minimize a risk-neutral objective on a class of policies $\Pi$, i.e.,

$$J(\pi) := \mathbb{E}_{\tau\sim\rho_{\pi,P}}\Big[R_T(\tau)\Big] \tag{4.1}$$

The risk-sensitive exponential criterion with a risk parameter $\beta$ is given by

$$J_\beta(\pi) := \frac{1}{\beta}\log \mathbb{E}_{\tau\sim\rho_{\pi,P}}\Big[e^{\beta R_T(\tau)}\Big] \tag{4.2}$$

where $R_T(\tau) := \sum_{t=1}^{T}\gamma^t r(s_t, a_t)$ is the total discounted cost and $\mathbb{E}_{\tau\sim\rho_{\pi,P}}[\cdot]$ denotes the expectation taken over the system's trajectory generated by following the policy $\pi$ in an environment with transition kernel $P$. The parameter $\beta\in\mathbb{R}$ is a real constant parameter. A positive risk parameter

$\beta > 0$ corresponds to a risk-averse behavior and a negative risk parameter $\beta < 0$ corresponds to a risk-seeking behavior.

To provide guarantees, a robust MDP considers the case where the transition probability is determined in an adversarial way, that is, when action $a$ is taken at state $s$, the transition probability $p(\cdot|s,a)$ can be an arbitrary element of some uncertainty set $U(p(\cdot|s,a))$. Thus, the Distributionally Robust objective is given by

$$J_{DR}(\pi) := \min_{\hat{P} \in U(P)} \mathbb{E}_{\tau \sim \rho_{\pi,\hat{P}}} \Big[ R_T(\tau) \Big] \tag{4.3}$$

where $U(P)$ is the uncertainty set given the transition kernel $P$, which defines the uncertainty set for each state-action pair, i.e., $U(p(\cdot|s,a))$. For example, model-based RL approaches, which first learn the system model and then use the learned model to find the optimal policy, are bound to have model estimation errors when the model is learned from a finite number of samples. Note that the samples for the worst-case trajectories are not attainable by the agent. The KL-regularized RL objective [31, 32], which augments the risk-neutral RL objective with expected KL divergence between the policy and a reference policy $\pi_0$ (parametrized by $\theta_0$) over the system's trajectory is given by

$$J^{\lambda}_{KL}(\theta, \theta_0) := \mathbb{E}_{\tau \sim \rho_\theta} \Big[ R(\tau) - \lambda D_{KL} \Big( \pi_\theta(\cdot|s_t), \pi_{\theta_0}(\cdot|s_t) \Big) \Big] \tag{4.4}$$

where the expectation is taken under policy's trajectory distribution (parameterized by $\theta$), i.e.,

$s_1 \sim p_1$, $a_t \sim \pi_\theta(\cdot|s_t)$ and $s_{t+1} \sim p_{s_{t+1},s_t}(a_t)$, and

$$D_{KL}(Q, P) = \begin{cases} \mathbb{E}_Q\left[\log \frac{dQ}{dP}\right] & \text{if } Q \ll P \\ \infty & \text{otherwise} \end{cases}$$

is the Kullback–Leibler (KL) divergence between the two probability distribution $Q$ from $P$.

The regularization weight $\lambda$ is a real value non-negative constant. The weight $\lambda$ is a design parameter that controls the level of regularization. In a more precise view, Eq. (4.4) represents a "scalarization" approach to the trade-off between two performance metrics. This view has not been studied in the literature. The reference parameter $\theta_0$ maybe given by an expert, or in the absence of a close-to-optimal reference policy, a host of RL algorithms, such as TRPO [39] and MPO [40], have adopted an iterative scheme, in which at each time-step, the reference policy parameter is fixed to the policy parameter obtained from the previous time-step, starting from some initial reference policy parameter, and then optimizing the KL-regularized objective over the policy parameter; repeating the two steps until convergence.

The KL-divergence regularization penalizes the distance from the reference policy. Thus, the KL-regularized objective of Eq. (4.4) maximizes the reward obtained over the system's trajectory while also stays close to a reference behaviour characterized by the reference policy $\pi_{\theta_0}$.

Note that for a choice of uniform distribution as the reference policy, the KL-regularized objective is equivalent to the maximum entropy objective [32] (up to a constant). Therefore, we only consider the more general case of KL-regularized objective from hereon.

Risk-sensitivity, a concept deeply rooted in the pursuit of mitigating uncertainties, enables us to consider the tails of probability distributions and the potential impact of rare yet critical

events. With risk-sensitivity, we can craft strategies that not only optimize for the expected outcomes but also account for the unexpected and extreme scenarios that may arise. Robustness, on the other hand, embodies the desire for reliability and adaptability. It is an acknowledgment that our models and algorithms may be imperfect, and uncertainties may manifest differently than expected. To embrace robustness is to prepare for the unknown by designing systems that can withstand perturbations, handle data anomalies, and gracefully adjust to unforeseen circumstances. Regularization, by imposing constraints and biases, balances between exploiting available data and avoiding overfitting. It prevents our models from becoming excessively specialized to the training data. In this exploration, we embark on a journey to unravel the intricate tapestry woven by risk-sensitivity, robustness, and regularization. We seek to comprehend their intertwined nature and uncover the delicate trade-offs and synergies they entail. Through this journey, we aspire to glean a deeper understanding of how these concepts shape decision-making, optimization, and the emergent behavior of agents in a multitude of domains.

## 4.2 Related Work

Decades of work on risk-sensitive optimization and control, starting with the mean-variance formulation in Portfolio Theory [66], Linear-Quadratic-Exponential-Gaussian (LEQG) [43] in Linear (risk-sensitive) Control Theory and risk-sensitive MDP in the theory of Markov Decision Processes [51], and empirical economic models, such Prospect Theory (PT) [3] and Cumulative Prospect Theory (CPT) [4], have well-motivated the research in risk-sensitive Reinforcement Learning (RL).

Jacobson [43] introduced the risk-sensitive exponential of integral criterion and solved the

optimal state feedback controller in the discrete-time linear-Gaussian systems with Quadratic cost context (LEQG), which turned out to be a linear function of the state, and further established an equivalence with a deterministic dynamic game [43]. Speyer et al. [52] solved the more general LEQG with an output feedback controller. The connection between risk-sensitive (exponential of integral) control and Robust control and their dynamic game formulation, starting with Glover and Doyle [67] for linear systems, has been long established for general non-linear systems (both finite and infinite time horizon), and general solution has been obtained (via the large deviation limit as noise in stochastic problem tends to zero), which is expressed as a coupled Ricatti equations, one forward in time and the other backward in time [54]; By taking the small noise (high-risk attitude) limit, a deterministic dynamic game which is closely related to the robust control problem is obtained. These results established that the exponential of integral criterion does not need to be postulated, but instead it emerges naturally from the mathematical analysis of robust output feedback control, which in turn suggests that the risk-sensitive control exhibits robustness.

The regularized RL objectives such as the maximum entropy [33, 34, 35, 36] and Kullback–Leibler (KL)-divergence regularization [31, 32] attempt to mitigate the issues associated with risk-neutral RL objectives by augmenting the risk-neutral objective with a regularization term. The maximum entropy augments the risk-neutral objective with the Shannon entropy of the policy, and the KL-regularized objective uses a KL divergence term between the policy and a reference policy. Maximizing the maximum-entropy objective is equivalent to the maximization of the KL-regularized objective with a uniform distribution as the reference policy [32]. KL regularization has shown promising results and is at the heart of algorithms, such as Trust Region Policy Optimization (TRPO) [39] and Maximum A Posteriori Policy Optimization (MPO) [40].

The risk-sensitive RL objectives, that is, the objective functions that incorporate some notion of risk, e.g., higher moments of return as an example, into the objective function have shown promising results in addressing some of the issues associated with risk-neutral RL [24, 25, 26, 27, 28]. There has been a small, albeit growing, number of results on risk-sensitive RL, considering different risk-sensitive objectives, such as the conditional value at risk [24, 25] and the variance [23, 26]. One particular such risk-sensitive objective is the exponential criterion [43] which has been the cornerstone of risk-sensitive control literature and has been substantially studied over the last five decades [17, 43, 52, 53, 54, 55, 56, 57]. The exponential criterion is a mathematically convenient and intuitively appealing risk measure with a firm theoretical foundations rooted in Large Deviation Theory. Most importantly, in our prior work [17, 54, 55, 56, 57], we established that the exponential of an integral criterion emerges from the mathematical analysis of the so called "four block" or H-infinity output robust control in its full generality. Further analytical development of such mathematical analysis to the problems investigated in the present paper will be presented elsewhere.

## 4.3 Risk-Sensitivity and Distributionally Robust RL

We explore the relation between the risk-sensitive exponential and Distributionally Robust RL objectives, and in doing so, we unify some of the popular Reinforcement Learning algorithms. Such equivalence (I) allows to understand a number of well-known Reinforcement Learning algorithms from a risk minimization perspective and (II) establishes the robustness properties of risk-sensitive exponential objective in the RL context, which in turn provides a theoretical justification for the robust performance of risk-sensitive RL algorithms in the literature. The robustness

of exponential criteria motivates risk-sensitizing current risk-neutral RL algorithms using such criteria.

### 4.3.1 Coherent Exponential criterion

We connect the Distributionally Robust objective to a coherent version of the exponential criteria which has an equivalent optimization over policy to the exponential criteria. Recall that the exponential criterion $J_\beta$ (cf. Eq. (3.2)) is convex but not coherent. It can be made coherent by enforcing a constraint. To that end, let's construct an adjusted exponential criterion $J_c^{\beta,\alpha}(\pi)$ given by

$$J_c^{\beta,\alpha}(\pi) := J_\beta(\pi) - \frac{1}{|\beta|}\ln(\alpha) \qquad 0 < \alpha < 1$$

where $J_\beta(\pi) := \frac{1}{\beta}\log \mathbb{E}_{\tau\sim\rho_{\pi,P}}\left[e^{\beta R_T(\tau)}\right]$ is the (convex) exponential criterion (cf. Eq. (3.2)), and $\alpha$ is a constant parameter.

Note that the adjustment term $\frac{1}{\beta}\ln(\alpha)$ is independent of the policy. Thus, the optimal policy with respect to the adjusted exponential criterion $J_c^{\beta,\alpha}(\pi)$ and the exponential criterion $J_\beta(\pi)$ are the same. That is to say, the optimization of the two objectives are equivalent.

The supremum of the adjusted exponential criterion over the risk aversion parameter $\beta < 0$ is called Entropic Value at Risk [68] with confidence '$1-\alpha$', i.e.,

$$J_{c^-}^{\alpha}(\pi) := \sup_{\beta<0} J_c^{\beta,\alpha}(\pi)$$

The Entropic Value at risk is a coherent risk measure and as such has a dual representation

given by [68]

$$J_{c^-}^{\alpha}(\pi) := \inf_{\substack{\hat{\rho} \\ \hat{\rho} \in \{\hat{\rho} \ll \rho : D(\hat{\rho}, \rho) < -\ln \alpha\}}} \mathbb{E}_{\tau \sim \hat{\rho}} \Big[ R_T(\tau) \Big] \tag{4.5}$$

**Remark 3.** *It is evident from Eq. (4.5) that the KL-regularized and KL-constrained algorithms such TRPO [39] and PPO [38] are attempts to iteratively optimize the convex and coherent risk-sensitive criterion.*

In a similar manner, it is straightforward to show that for a given $\beta < 0$ (risk-seeking) the coherent exponential criterion is equivalent to

$$J_{c^+}^{\alpha}(\pi) = \sup_{\substack{\hat{\rho} \\ \hat{\rho} \in \{\hat{\rho} \ll \rho : D(\hat{\rho}, \rho) < -\ln \alpha\}}} \mathbb{E}_{\tau \sim \rho_{\pi,P}} \Big[ R_T(\tau) \Big] \tag{4.6}$$

We use $J_c^{\alpha}$ to refer to both $J_{c^-}^{\alpha}$ and $J_{c^+}^{\alpha}$ at the same time. We call $J_c^{\alpha}$ the coherent exponential criterion at level c [69].

The relation between the (convex) exponential criterion and coherent exponential criterion for a positive risk parameter $\beta > 0$ is given by [69]

$$J_{c^+}^{\alpha}(\pi) = \max_{\beta > 0} \left\{ J_{\beta}(\pi) - \frac{\ln \alpha}{\beta} \right\}$$

and for a negative risk parameter $\beta < 0$ is given by

$$J_{c^-}^{\alpha}(\pi) = \max_{\beta < 0} \left\{ J_{\beta}(\pi) + \frac{\ln \alpha}{\beta} \right\}$$

### 4.3.2 Distributionally Robust

We show that the risk-averse agent has a distributionally robust objective with a certain uncertainty set. Recall that when the model of the system, i.e., the transition kernel $P$, is known up to some uncertainty set $\mathcal{U}(P)$ and the objective is to minimize the total cost over the system's trajectory, the Distributionally Robust (DR) Objective (cf. Eq. (4.3)), that is, the worst-case Criteria with respect to the uncertainty set, is given by

$$J_{DR}(\pi) := \min_{\hat{P}\in\mathcal{U}(P)} \mathbb{E}_{\tau\sim\rho_{\pi,\hat{P}}}\Big[R_T(\tau)\Big]$$

where $R_T(\tau)=\sum_{t=1}^{T}\gamma^t r_t$ is the total cost over the system trajectory, and the expectation is taken with respect to the policy $\pi$ and the transition kernel $\hat{P}$. We formally state this connection in the following theorem.

**Theorem 3.** *The coherent risk-sensitive exponential criterion with a positive risk parameter $\beta<0$ (risk-aversion) is equivalent to a distributionally robust objective with the uncertainty set U(P) given by Eq. (4.7). That is to say,*

$$J^{\alpha}_{c-} := \min_{\hat{P}\in\mathcal{U}(P)} \mathbb{E}_{\tau\sim\rho_{\pi,\hat{P}}}\Big[R_T(\tau)\Big]$$

*Furthermore, for a given negative risk-parameter $\beta>0$ (risk-seeking)*

$$J^{\alpha}_{c+} = \max_{\hat{P}\in\mathcal{U}(P)} \mathbb{E}_{\tau\sim\rho_{\pi,\hat{P}}}\Big[R_T(\tau)\Big]$$

*where $J^{\alpha}_{c+}$ and $J^{\alpha}_{c-}$ are the coherent risk-sensitive exponential criterion for $\beta>0$ and $\beta<0$, respec-*

*tively.*

*Proof.* By lemma 3 in [62], given a policy $\pi$, we have

$$D(\rho_{\pi,\hat{p}}, \rho_{\pi,p}) = T\mathbb{E}_{\tau\sim\rho_{\pi,p}}\left[D\left(\hat{p}(\cdot|s,a), p(\cdot|s,a)\right)\right]$$

Then, using the dual representation of the coherent risk measure $J_c^\alpha$ (cf. Eq. (4.5) and (4.6)), we conclude that the coherent exponential criterion for $\beta>0$ (risk-aversion) is a Distributionally Robust objective with a certain uncertainty set, i.e.,

$$U(P) = \left\{\hat{P} : \mathbb{E}_{\tau\sim\rho_{\pi,P}}\left[D\left(\hat{p}, p\right)\right] \leq \frac{-\ln\alpha}{T}\right\} \tag{4.7}$$

□

**Remark 4.** *Theorem 3 states the relation between the coherent risk-sensitive exponential criterion and the distrubutionally robust RL objectives, and by doing so, establishes the robustness property of risk-sensitive exponential criterion.*

**Remark 5.** *It is clear that the Uncertainty set* $U'(P)$ *given by*

$$U'(P) = \left\{\hat{P} : D(\hat{p}(\cdot|s,a), p(\cdot|s,a)) \leq \frac{-\ln\alpha}{T} \quad \forall(s,a)\right\}$$

*is a subset of the uncertainty set* $U(P)$*. Thus,*

$$J_{c+}^\alpha(\pi) \geq \max_{\hat{P}\in U'(P)} \mathbb{E}_{\tau\sim\rho_{\pi,\hat{p}}}\left[R_T(\tau)\right], \qquad J_{c-}^\alpha(\pi) \leq \min_{\hat{P}\in U'(P)} \mathbb{E}_{\tau\sim\rho_{\pi,\hat{p}}}\left[R_T(\tau)\right]$$

## 4.4 Risk-Sensitivity and Regularized RL

### 4.4.1 KL-Divergence Regularized RL

we analytically explore the connection between the regularized and the risk-sensitive RL using the MDP framework. Before we proceed with chapter, we first define the Maximum Entropy and KL regularized objectives. Maximum Entropy is another popular RL objective, which augments the risk-neutral RL objective with an entropy regularization, i.e.,

$$J_{ent}(\theta) := \mathbb{E}_{\tau\sim\rho_\theta}\Big[R(\tau)\Big] + \lambda\mathbb{E}_{s_t\sim p(s_{t+1}|s_t,a_t)}\Big[\sum_{t=1}^{T}\mathcal{H}^{\pi}(\cdot|s_t)\Big] \tag{4.8}$$

where $\mathcal{H}^{\pi}(\cdot|s_t)=-\mathbb{E}_{a\sim\pi(a_t|s_t;\theta)}\Big[\log\pi(a_t|s_t;\theta)\Big]$ is the entropy of policy $\pi$ in state $s_t$, and the regularization weight $\lambda$ is a real value non-negative constant. The weight $\lambda$ is a design parameter that controls the level of regularization. Maximum-entropy RL objective helps with exploration, prevents pre-mature convergence to sub-optimal policies, and provides better generalization, which leads to more robust policies.

Recall that the KL-regularized RL objective [31, 32], which augments the risk-neutral RL objective with expected KL divergence between the policy and a reference policy $\pi_0$ (parametrized by $\theta_0$) over the system's trajectory is given by

$$J^{\lambda}_{KL}(\theta,\theta_0) := \mathbb{E}_{\tau\sim\rho_\theta}\Big[R(\tau) - \lambda D_{KL}\Big(\pi_\theta(\cdot|s_t),\pi_{\theta_0}(\cdot|s_t)\Big)\Big] \tag{4.9}$$

where the expectation is taken under policy's trajectory distribution (parameterized by $\theta$), i.e.,

$s_1 \sim p_1$, $a_t \sim \pi_\theta(\cdot|s_t)$ and $s_{t+1} \sim p_{s_{t+1},s_t}(a_t)$, and

$$D_{KL}(Q, P) = \begin{cases} \mathbb{E}_Q\left[\log \frac{dQ}{dP}\right] & \text{if } Q \ll P \\ \infty & \text{otherwise} \end{cases}$$

is the Kullback–Leibler (KL) divergence between the two probability distribution $Q$ from $P$.

The regularization weight $\lambda$ is a real value non-negative constant. The weight $\lambda$ is a design parameter that controls the level of regularization. In a more precise view, Eq. (4.4) represents a "scalarization" approach to the trade-off between two performance metrics. This view has not been studied in the literature. The reference parameter $\theta_0$ maybe given by an expert, or in the absence of a close-to-optimal reference policy, a host of RL algorithms, such as TRPO [39] and MPO [40], have adopted an iterative scheme, in which at each time-step, the reference policy parameter is fixed to the policy parameter obtained from the previous time-step, starting from some initial reference policy parameter, and then optimizing the KL-regularized objective over the policy parameter; repeating the two steps until convergence.

The KL-divergence regularization penalizes the distance from the reference policy. Thus, the KL-regularized objective of Eq. (4.4) maximizes the reward obtained over the system's trajectory while also stays close to a reference behaviour characterized by the reference policy $\pi_{\theta_0}$.

Note that for a choice of uniform distribution as the reference policy, the KL-regularized objective is equivalent to the maximum entropy objective [32] (up to a constant). Therefore, we only consider the more general case of KL-regularized objective from hereon.

#### 4.4.1.1 The Connection Between The KL-regularized Objective and The Risk-sensitive Exponential Criteria

Recall that by the dual representation theorem of Convex risk measures [49, 50], the exponential criteria has a dual representation given by [49]

$$J_\beta(\theta) = \sup_{\hat{\theta}} \left\{ \mathbb{E}_{\tau\sim\rho_{\hat{\theta}}}\Big[R(\tau)\Big] - \frac{1}{\beta} D_{KL}\Big(\rho_{\hat{\theta}}(\tau), \rho_\theta(\tau)\Big) \right\}$$

where the support of $\rho_{\hat{\theta}}$ is contained within the support of $\rho_\theta$, that is to say, $\rho_{\hat{\theta}}$ is absolutely continuous with respect to $\rho_\theta$. Note that the support of a probability distribution $\rho_\theta$ is defined as the set $\{\tau \in \mathcal{T} | \rho_\theta(\tau) > 0\}$.

The dual representation of the exponential criteria presented here is an application of the theorem of dual representation of Convex risk measures [49, 50] for the special case of entropic risk measure. For the general statement of the theorem, proof, and detailed explanation, please see [49, 50] and references therein.

By noting the definition of the trajectory distribution $\rho_\theta$ (cf. Eq (2.1)), It can be shown that, see Theorem 5 in [62],

$$D_{KL}\Big(\rho_{\hat{\theta}}(\tau), \rho_\theta(\tau)\Big) = T\mathbb{E}_{\tau\sim\rho_{\hat{\theta}}}\Big[D_{KL}\Big(\pi_{\hat{\theta}}(\cdot|s_t), \pi_\theta(\cdot|s_t)\Big)\Big]$$

where $T$ is the time horizon. Thus, we have

$$J_\beta(\theta) = \sup_{\hat{\theta}} \mathbb{E}_{\tau\sim\rho_{\hat{\theta}}}\Big[R(\tau) - \frac{T}{\beta} D_{KL}\Big(\pi_{\hat{\theta}}(\cdot|s_t), \pi_\theta(\cdot|s_t)\Big)\Big] \tag{4.10}$$

where the support of $\pi_{\hat{\theta}}$ is contained within the support of $\pi_\theta$. Note that such condition can be satisfied, for example, if the policies are always non-zero for all actions given a state.

Recall $J^{\lambda}_{KL}(\hat{\theta}, \theta) = \mathbb{E}_{\tau\sim\rho_{\hat{\theta}}}\left[R(\tau) - \lambda D_{KL}\left(\pi_{\hat{\theta}}(\cdot|s_t), \pi_\theta(\cdot|s_t)\right)\right]$ (cf. Eq. (4.4)). Thus, the exponential criteria with a risk parameter $\beta>0$ and for a given policy parameter $\theta$ is equal to the supremum of the KL-regularized objective for a given reference policy parameter $\theta$ and the regularization weight $\lambda=T/\beta$, that is,

$$J_\beta(\theta) = \sup_{\hat{\theta}} J^{\frac{T}{\beta}}_{KL}(\hat{\theta}, \theta) \tag{4.11}$$

For notational simplicity, from hereon, without loss of generality, we assume that the supremum can be attained and we use a max operator instead of sup operator. Our results hold for sup operator as well.

The dual representation of the exponential criteria (cf. Eq. (4.11)) reveals the relationship between the exponential criteria (cf. Eq. (3.1)) and the KL-regularized RL objective (cf. Eq. (4.4)). We formally state the relationship between the exponential criteria and the KL-regularized RL objective in the following theorem.

**Theorem 4.** *The maximization of the exponential criteria $J_\beta(\theta)$ (cf. Eq (3.1)) with a positive risk parameter $\beta>0$ is equivalent to the maximization of the KL-regularized objective $J_{KL}(\hat{\theta}, \theta)$ (cf. Eq. (4.4)) jointly over the policy parameters $\hat{\theta}$ and the reference policy parameters $\theta$, that is,*

$$\arg\max_{\theta} J_\beta(\theta) = \arg\max_{\hat{\theta},\theta} J^{\frac{T}{\beta}}_{KL}(\hat{\theta}, \theta)$$

*where*

$$J_{KL}^{\frac{T}{\beta}}(\hat{\theta}, \theta) = \mathbb{E}_{\tau \sim \rho_{\hat{\theta}}}\left[R(\tau) - \frac{T}{\beta} D_{KL}\left(\pi_{\hat{\theta}}(\cdot|s_t), \pi_{\theta}(\cdot|s_t)\right)\right]$$

*is the KL-regularized objective with reference policy parameter* $\theta$ *and the regularization weight* $T/\beta$ *with* $T$ *being the time horizon.*

*Proof.* It follows straightforwardly from the dual representation of exponential criteria (cf. Eq (3.7)) and the definition of KL-regularized objective (cf. Eq (4.4)). □

**Remark 6.** *Theorem 4 suggests that the risk-sensitive exponential criteria has an inherent mechanism for considering the optimal reference policy in the KL-regularized objective.*

**Remark 7.** *Solving an optimization problem over two sets of disjoint variables, such as the optimization of the KL-regularized objective jointly over the policy parameter* $\hat{\theta}$ *and the reference policy parameter* $\theta$ *(cf. Theorem 4), can be attempted using a multitude of optimization methods. Alternating Optimization (AO) [70] methods, e.g., Alternating Gradient Descent (A-GD), which is a simple and popular algorithm, provide one prospect for solving such joint optimization problems. Alternating Optimization methods are iterative procedures for maximizing (or minimizing) a multi-variable function jointly over all variables by alternating restricted maximization over the individual subsets of variables— that is to say, the joint optimization problem given in Theorem 4 can be solved by fixing the reference policy parameter obtained from the previous iteration, staring from some initial value, optimizing over the policy parameter, and then fixing the policy parameter with the value obtained and optimizing over the reference policy parameter; repeating*

*these two steps until a stopping condition is met. That is,*

$$\hat{\theta}_{t+1} = \arg\max_{\hat{\theta}} J_{KL}^{\frac{T}{\beta}}(\hat{\theta}, \theta_t), \quad \theta_{t+1} = \arg\max_{\theta} J_{KL}^{\frac{T}{\beta}}(\hat{\theta}_{t+1}, \theta)$$

A similar iterative optimization procedure for the class of Path Integral control problems, a restricted class of non-liner control problems with arbitrary dynamics and state cost, but with a linear dependence of the control on the dynamics and quadratic control cost, and for the special case of linearly parameterized policies, have been suggested in [71].

#### 4.4.1.2 Iterative Policy Optimization Based on KL-regularized Objective and Its Connections to Risk-sensitive RL with Exponential Criteria

Although Alternating Optimization algorithms (c.f. Remark 7) have not been well explored to develop RL algorithms and is the subject of our future work, similar iterative algorithms based on the KL-regularized objective are at the core of well-known RL algorithms, such as Trust Region Policy Optimization (TRPO) [39] and Maximum A Posterior Optimization (MPO) [40], which consist of an iterative procedure in which at each time-step, the reference policy parameter is fixed to the policy parameter obtained from the previous time-step, starting from some initial reference policy parameter, and then optimizing the KL-regularized objective over the policy parameter; repeating the two steps until convergence. Note that this iterative procedure is different from the Alternating Optimization procedure described in Remark 7.

We show that this iterative optimization procedure is an attempt to approximately maximize the risk-sensitive exponential criteria. We first formally state the relation between the maximization of the risk-sensitive exponential criteria and the iterative optimization procedure in

the following theorem and give the proof immediately after. Then, we offer an interpretation of the iterative optimization procedure as the use of Minorization-Maximization (MM).

**Theorem 5.** *The following iterative optimization scheme is an iterative attempt to maximize the exponential criteria with a positive risk parameter* $\beta > 0$*,*

$$\theta_{t+1} = \arg\max_{\hat{\theta}} J_{KL}^{\frac{T}{\beta}}(\hat{\theta}, \theta_t)$$

*where*

$$J_{KL}(\hat{\theta}, \theta_t) = \mathbb{E}_{\tau \sim \rho_{\hat{\theta}}}\left[R(\tau) - \frac{T}{\beta} D_{KL}\left(\pi_{\hat{\theta}}(\cdot|s_t), \pi_{\theta_t}(\cdot|s_t)\right)\right]$$

*(cf. Eq.* (4.4)*), is the KL-regularized objective with the regularization weight* $T/\beta$ *and the reference policy parameter* $\theta_t$ *obtained from the previous time-step, starting with an arbitrary initial parameter* $\theta_0$*.*

*Proof.* We first show that at each time-step, the iterative procedure described in Theorem 5 generates a parameter policy for which the exponential criteria has a value at least as high as the

policy parameter in the previous time-step. To see that, by using Eq. (4.11), we have

$$
\begin{aligned}
J_\beta(\theta_{t+1}) &= \max_{\hat{\theta}} J^{\frac{T}{\beta}}_{KL}(\hat{\theta}, \theta_{t+1}) \\
&\geq J^{\frac{T}{\beta}}_{KL}(\theta_{t+1}, \theta_{t+1}) \\
&= \mathbb{E}_{\tau\sim\rho_{\theta_{t+1}}}\Big[R(\tau)\Big] \\
&\geq \mathbb{E}_{\tau\sim\rho_{\theta_{t+1}}}\Big[R(\tau) - \frac{T}{\beta} D_{KL}\Big(\pi_{\theta_{t+1}}(\cdot|s_t), \pi_{\theta_t}(\cdot|s_t)\Big)\Big] \\
&= J^{\frac{T}{\beta}}_{KL}(\theta_{t+1}, \theta_t) \\
&= J_\beta(\theta_t)
\end{aligned}
$$

The second line follows from the definition of sup operator, that is, the supremum is greater than or equal to the value of the function for any value of the decision variable, e.g. $\theta_{t+1}$. The third line is straightforward use of Eq. (4.4) by noting that $D_{KL}\Big(\pi_{\theta_{t+1}}(\cdot|s_t), \pi_{\theta_{t+1}}(\cdot|s_t)\Big)=0$. The forth line follows directly from the non-negativity of KL divergence. The fifth line follows from the definition of the KL-regularized objective (cf. Eq. (4.4)). The last line follows from the iterative procedure that generates $\theta_{t+1}$ by noting that $\theta_{t+1} = \arg\max_{\hat{\theta}} J_{KL}(\hat{\theta}, \theta_t)$, that is to say,

$$
J^{\frac{T}{\beta}}_{KL}(\theta_{t+1}, \theta_t) = \arg\max_{\hat{\theta}} J^{\frac{T}{\beta}}_{KL}(\hat{\theta}, \theta_t) = J_\beta(\theta_t)
$$

Thus, we have shown that at each iteration of the iterative scheme the value of the exponential criteria is increased. $J_\beta(\theta)$ will converge to a local optimum or a saddle point as $t$ goes to infinity. □

Then, it is easy to see that the KL iterative optimization procedure can be thought of as

the Minorization-maximzaition (MM) algorithm for optimizing the risk-neutral expected cumulative reward. The KL-regularized objective $J_{KL}^{\frac{T}{\beta}}(\hat{\theta}, \theta_t)$ (cf. Eq (4.4)) minorizes the risk-neutral expected cumulative reward $J(\theta)$ (cf Eq. (2.2)), that is, it is tangent to the expected cumulative reward at a given policy parameter $\theta_t$ and it is dominated by the expected cumulative reward at all points, i.e.,

$$J(\theta_t) = J_{KL}^{\frac{T}{\beta}}(\theta_t, \theta_t), \quad J(\theta) \geq J_{KL}^{\frac{T}{\beta}}(\theta, \theta_t) \qquad \forall \theta \in \mathbb{R}^d$$

where $\theta_{t+1} = \arg\max_{\hat{\theta}} J_{KL}^{\frac{T}{\beta}}(\hat{\theta}, \theta_t)$

To see the connection between the described iterative optimization procedure and the Minorization-Maximization scheme, recall the KL-regularized objective (cf. Eq. (4.4)). By non-negativity of KL divergence, we can see that the risk-neutral expected cumulative reward at the point $\theta_t$, $J(\theta_t)$, is lower bounded by the KL-regularized objective with any regularization weight, e.g. $T/\beta$, and any reference policy parameter $\theta$. By noting that $D_{KL}\left(\pi_{\theta_t}(\cdot|s_t), \pi_{\theta_t}(\cdot|s_t)\right) = 0$, we can see the lower bound is tight when the policy parameter $\hat{\theta}$ is equal to the reference policy parameter.

Thus, the KL-regularized objective can be treated as a surrogate function to the risk-neutral expected cumulative reward. To maximize the expected cumulative reward $J_\beta(\theta)$, one can maximize the surrogate function for $\theta$ to generate the next iterate, starting from some initial value,

$$\theta_{t+1} = \arg\max_{\hat{\theta}} J_{KL}^{\frac{T}{\beta}}(\hat{\theta}, \theta_t)$$

### 4.4.2 (h,f)-Divergence Regularized RL

Dual representation of Convex risk measures [49, 50] states that any Convex risk measure $J(\theta)$ on $\mathcal{T}$ can be expressed as,

$$J(R;\theta) = \sup_{\hat{\theta}} \left\{ \mathbb{E}_{\tau\sim\rho_{\hat{\theta}}}\Big[R\Big] - D(\rho_{\hat{\theta}}) \right\} \tag{4.12}$$

where $\rho_{\hat{\theta}}$ is absolutely continuous with respect to $\rho_\theta$ and $D$ is the penalty function, a function from the set all absolutely continuous probability distributions with respect to $\rho_\theta$ to the reals $\mathbb{R}$.

#### 4.4.2.1 $(h, f)$-Divergence and Risk measures

As it is noted in [72], an extension of Entropic risk measure to a broader class of Convex risk measures is obtained by substituting the penalty term $\alpha(\cdot)$ in the dual representation of Convex risk measures (c.f. Eq (4.12)) with a convex $(h, f)$-divergence, that is,

$$J_f^h(\theta) = \sup_{\tau\sim\rho_{\hat{\theta}}} \left\{ \mathbb{E}_{\tau\sim\rho_{\hat{\theta}}}\Big[R(\tau)\Big] - D_f^h\Big(\rho_{\hat{\theta}}, \rho_\theta\Big) \right\} \tag{4.13}$$

where $D_f^h$ is a convex (h,f)-divergence, an extension of the f-divergence [73, 74, 75, 76], defined as

$$D_f^h\Big(\rho_{\hat{\theta}}, \rho_\theta\Big) = \sum_{i=1}^{I} w_i h_i\Big(\int f(\frac{d\rho_{\hat{\theta}}}{d\rho_\theta}) d\rho_\theta\Big)$$

for probabilities $\rho_{\hat{\theta}}$ and $\rho_\theta$ such that $\rho_\theta$ is absolutely continuous with respect to $\rho_{\hat{\theta}}$ and where the distortion functions $h_i$ is a non-decreasing and continuous function with $h_i(0)=0$, $w_i$ is a

positive weight, and the function $f_i$ satisfies the conditions for a f-divergence risk measure. (h, f)-divergence risk measure is a convex measure of risk. It should be noted that (h,f)-divergences are equal to zero if and only if the two distributions are the same.

Note that f-divergences are (h,f)-divergences with an identity distortion function $h$, i.e. $h_i(x)=x$ and weight $w_i=1$. KL-divergence is a f-divergence with $f(x)=x\ln x$. For two probability distributions $\rho_\theta$ and $\rho_{\hat{\theta}}$ over the same space, the f-divergence $\rho_\theta$ from $\rho_{\hat{\theta}}$ is defined as

$$D_f(\rho_{\hat{\theta}}, \rho_\theta) = \int f(\frac{d\rho_{\hat{\theta}}}{d\rho_\theta}) d\rho_\theta$$

where f is a convex function on $(0, \infty)$ such that $f(1)=0$. KL-divergence, reverse KL divergence, total variation distance, Rényi divergence of order $\alpha$ ($\alpha$-divergence), Hellinger distance, Pearson $\chi^2$-divergence, Neyman (reverse Pearson) $\chi^2$-divergence, and Jensen-Shannon divergence are instances of f-divergence.

**Remark.** Such f-divergence regularized objectives have been studied in the RL literature [77, 78]. The risk optimization perspective unifies such algorithms under the same banner and provides a systematic mechanism for further algorithmic development.

**Remark.** [77] explores an algorithmic approach for $\alpha$-divergence constrained policy improvement. However, it seems the algorithm is motivated as a generalization of KL-regularized constrained algorithms to f-divergence. Our discussion provides a theoretical explanation of the choice of f-divergence and constraint between successive policies during parametric policy iteration.

#### 4.4.2.2 Iterative Policy Optimization Based on (h,f)-regularized Objective and Its Connections to Risk-sensitive RL with Exponential Criteria

RL agents seek to find a set of policy parameters $\theta^*$ so as to optimize the system performance measure, that is, RL-agents aim to solve the following optimization problem

$$\theta^* = \arg\max_{\theta} J(\theta) \tag{4.14}$$

where $J(\theta)$ is the system performance measure which is a function of the agent's policy $\pi$ and hence the policy parameters $\theta$, e.g., $J^{\lambda}_{KL}$(c.f.Eq (4.4)), $J_{neu}$(c.f. Eq (2.2)), and $J_{ent}$(c.f. Eq (3.1)).

For the class of (h,f)-divergence based risk measures (c.f. Eq (4.13)), we have

$$\max_{\theta,\hat{\theta}} J^{h}_{f}(\hat{\theta}, \theta) := \max_{\hat{\theta}} \mathbb{E}_{\tau\sim\rho_{\hat{\theta}}}\Big[R(\tau)\Big] - D^{h}_{f}\Big(\rho_{\hat{\theta}}, \rho_{\theta}\Big) \tag{4.15}$$

Without loss of generality and for notational convenience, we assume the supremum is attainable and we substitute the supremum with a maximum from hereon.

An (h,f)-regularized objective is,

$$\bar{J}^{h}_{f}(\hat{\theta}, \theta) := \mathbb{E}_{\tau\sim\rho_{\hat{\theta}}}\Big[R(\tau)\Big] - D^{h}_{f}\Big(\rho_{\hat{\theta}}, \rho_{\theta}\Big) \tag{4.16}$$

It should be noted that the (h,f)-divergence regularized objective is a lower bound on the corresponding (h,f)-divergence risk measure.

We present a general iterative framework for solving such optimization problems (c.f. Eq (4.15)) using Alternating Optimization. We also show that such regularized objectives provide

a surrogate function for optimizing the risk-neutral objective in an iterative Minorize-Maximization (MM) approach.

**Alternation Optimization Algorithms**: The (h,f)-regularized optimization problem of Eq (4.15) is over two sets of disjoint variables, the policy parameter $\hat{\theta}$ and the reference policy parameter $\theta$. Such optimization problems can be attempted using a multitude of optimization methods. Alternating Optimization (AO) [70] methods, e.g., Alternating Gradient Descent (A-GD), provide one prospect for solving such joint optimization problems. Alternating Optimization methods are iterative procedures for maximizing (or minimizing) a multi-variable function jointly over all variables by alternating restricted maximization over the individual subsets of variables— fixing the reference policy parameter obtained from the previous iteration, starting from some initial value, optimizing over the policy parameter, and then fixing the policy parameter with the value obtained and optimizing over the reference policy parameter; repeating these two steps until a stopping condition is met. That is,

$$\hat{\theta}_{t+1} = \arg\max_{\hat{\theta}} J_f^h(\hat{\theta}, \theta_t)$$

$$\theta_{t+1} = \arg\max_{\theta} J_f^h(\hat{\theta}_{t+1}, \theta)$$

It should be noted that the iteration above is a blueprint for algorithms, not an algorithm. For example, a similar iterative optimization procedure for the class of Path Integral control problems, a restricted class of non-linear control problems with arbitrary dynamics and state cost, but with a linear dependence of the control on the dynamics and quadratic control cost, and for the special case of linearly parameterized policies, have been suggested in [71].

**Minorize-Maximization Algorithms:** MM algorithms construct a surrogate function that

is tangent to the function to be optimized at the current iterate and is dominated by it in all other points. Such surrogate function can be designed to be more amenable to optimization and more robust (e.g. less sensitive to outliers). We show that (h,f)-divergence regularized objective is a surrogate function for the risk-neutral objective. Recall the class of (h,f)-divergence risk measures (cf. Eq. (4.4)). By non-negativity of (h,f)-divergence, we can see that the risk-neutral expected cumulative reward at the point $\theta_t$, $J_{neu}(\theta_t)$, is lower bounded by the (h,f)-regularized objective with any reference policy parameter $\theta$. By noting that $D_f^h\left(\rho_{\theta_t}, \rho_{\theta_t}\right)=0$, we can see the lower bound is tight when the policy parameter $\hat{\theta}$ is equal to the reference policy parameter. The (h,f)-regularized objective $J_f^h(\hat{\theta}, \theta_t)$ minorizes the risk-neutral expected cumulative reward $J_{neu}(\theta)$ (cf Eq. (2.2)), that is, it is tangent to the expected cumulative reward at a given policy parameter $\theta_t$ and it is dominated by the expected cumulative reward at all points, i.e.,

$$J(\theta_t) = J_f^h(\theta_t, \theta_t), \qquad J_f^h(\theta) \geq J_f^h(\theta, \theta_t) \quad \forall \theta \in \mathbb{R}^d$$

where $\theta_{t+1} = \arg\max_{\hat{\theta}} J_f^h(\hat{\theta}, \theta_t)$.Thus, the (h,f)-regularized objective can be treated as a surrogate function to the risk-neutral expected cumulative reward. To maximize the expected cumulative reward $J_{neu}(\theta)$, one can maximize the surrogate function for $\theta$ to generate the next iterate, starting from some initial value,

$$\theta_{t+1} = \arg\max_{\hat{\theta}} J_f^h(\hat{\theta}, \theta_t)$$

Part II

# From Theory to Practice: Embodying Risk in Reinforcement Learning

Part (I) demonstrates the advantages of utilizing exponential criteria for the development of Reinforcement Learning (RL) algorithms. In this section, we focus on developing algorithms that effectively leverage these exponential criteria. In Chapter 5, we focus on developing risk-sensitive Reinforcement Learning (RL) algorithms within the framework of Markov Decision Processes (MDPs). However, in Chapter 6, we broaden our scope by exploring the application of the Probabilistic Graphical Models (PGM) framework for developing risk-sensitive algorithms. Within this context, we delve into the PGM framework and examine its connection with the MDP framework. Chapter 7, concludes this section by exploring the effects of risk-sensitivity on trust, collaboration, and cooperation in multi-agent systems.

# Chapter 5: Risk-Sensitive RL Algorithms Using MDP

## 5.1 Overview

We present the development of a risk-sensitive reinforcement learning (RL) algorithm designed specifically for exponential criteria. Our approach leverages the Markov Decision Process (MDP) framework to achieve this goal. One could utilize various techniques to optimize the exponential criteria. One such approach is Monte Carlo methods, which leverage the power of sampling to estimate expected rewards by averaging the returns obtained from multiple episodes. By simulating numerous interactions and collecting samples, Monte Carlo methods provide reliable estimates of the exponential criteria, enabling the RL algorithm to learn and adapt. Another technique commonly used in RL algorithms is Temporal Difference (TD) learning, which combines bootstrapping and sampling to update the value function incrementally. TD algorithms estimate the value of a state by incorporating the current reward and the estimated value of the subsequent state. By iteratively updating these value estimates, TD algorithms can optimize the exponential criteria over time. By leveraging techniques like Monte Carlo methods, TD learning, and deep reinforcement learning, these algorithms can effectively improve their performance over time and achieve optimal results.

## 5.2 Related Work

A number of risk-sensitive reinforcement learning approaches have been studied in recent years; from constructing constraint stochastic optimization problems [18, 19, 20] or approximately solving mini-max optimization problems [21], to investigating different statistical measures of the objective function [22, 23, 24, 25, 26, 27]. The latter approach often yields to more favorable algorithmic implementations, since the computational problems associated with constraint optimization, and the convergence problems associated with the existence of multiple Nash equilibria, are avoided. In particular, the algorithms in [24, 25, 28, 29] use the conditional value at risk for policy search and the algorithms in [22, 23, 26, 30] use variance as the desired risk measure. Along the same directions, risk-sensitive algorithms have been developed based on postulated regularized objectives, such as Kullback-Leibler (KL) regularization [31, 32] or entropy regularization [33, 34, 35, 36, 37], leading to policy search methods, such as PPO [38], TRPO [39] and MPO [40].

Although these are ad-hoc approaches developed by experimental observations, there is a duality connection between KL- and entropy-regularized objectives and entropic risk measures [11, 27, 41, 42], associated with exponential criteria of the form: $\max_{\pi\in\Pi} \frac{1}{\beta}\mathbb{E}_{x\sim\pi}\left[\exp(\beta R(x))\right]$. In addition to this connection, exponential criteria are well-understood in the context of risk-sensitive control [17, 43, 44], and can lead to appealing algorithmic implementations [12].

## 5.3 Policy Gradient Methods with Exponential Criteria

Risk-neutral policy gradient methods for reinforcement learning have been extensively studied [79, 80, 81]. In this section, we present a brief overview of the most commonly used policy gradient RL algorithms and study their risk-sensitive counterpart based on the exponential criteria described in Chapter 3. In particular, we explore the properties of the risk-sensitive REINFORCE algorithm [12] and provide new analytic results for its implementation regarding its update rule and the convergence of its parameters.

### 5.3.1 Policy Gradient Methods

Policy Gradient (PG) methods are a class of Policy Search methods that use gradient ascend/descend schemes to search for the optimal policy [82, 83, 84]. The well-known REINFORCE [33] and Actor-Critic [85] algorithms are examples of Policy Gradient algorithms, which are particularly suitable for continuous action spaces. On-policy Monte Carlo policy gradient algorithms, e.g., REINFORCE [86], use episode samples to estimate the gradient. That is, at each iteration of the algorithm $t$, the parameters of the policy are updated using the following update rule

$$\theta_{t+1} = \theta_t + \alpha\widehat{\nabla J(\theta_t)} \tag{5.1}$$

where $\alpha \in \mathbb{R}$ is a constant step-size, i.e., learning rate, and $\widehat{\nabla J(\theta_t)} \in \mathbb{R}^d$ is an unbiased estimate of the gradient with respect to the policy parameter $\theta$. Monte-Carlo policy gradient estimates are unbiased estimates of the gradient of the system's performance measure, but suffer from high

variance. Such high variance estimates of the gradient lead to high sample complexity and hinder learning. This may limit their applications, especially in real-world scenarios where collecting a large number of samples may be expensive, time-consuming, or complex.

There has been numerous variance reduction techniques for policy gradient methods to mitigate the high variance of Monte-Carlo policy gradient estimates while preserving the stability and convergence properties of on-policy Monte Carlo policy gradient algorithms. These techniques attempt to reduce the variance of the gradient without introducing bias. The subtraction of an appropriately chosen baseline, both state-dependent [6, 86] and action-dependent [84, 87, 88, 89, 90] baselines, for variance reduction in policy gradient have been studied extensively over the last two decades. The use of action-dependent baselines and their effectiveness in reducing the variance over state-dependent baselines have been subject of much debate [91], so we refrain from considering action-dependent baselines in this work. Although an optimal state-dependent baseline exists [63], it is typically hard to find.

### 5.3.2 REINFORCE: A Monte Carlo Policy Gradient Method

An estimate of the gradient of the (risk-neutral) objective of (2.2) with respect to the policy parameters can be obtained using the policy gradient theorem [92], that is,

$$\nabla J(\theta) \propto \mathbb{E}_{\tau\sim\rho_\theta}\left[R(\tau)\sum_{t=0}^{|\tau|-1}\nabla \log \pi_\theta(a_t|s_t;\theta)\right] \tag{5.2}$$

Policy Gradient theorem suggests that the gradient estimate in Eq. (5.1) can be computed by Monte Carlo estimation of the expectation in Eq. (5.2). It is known that such estimation suffers from high variance [6]. To reduce variance, by taking advantage of the temporal structure

of the problem and causality, it has be shown that the gradient (expected value in Eq (5.2)) could be re-written in terms of reward-to-go $R_t := \sum_{t'=t}^{|\tau|-1} \gamma^{t'-t} r(s_{t'}, a_{t'})$ as follows:

$$\nabla J(\theta) \propto \mathbb{E}_{\tau \sim \rho_\theta} \Big[ \sum_{t=0}^{|\tau|-1} R_t \nabla \log \pi_\theta(a_t|s_t;\theta) \Big] \tag{5.3}$$

Using Eq. (5.3), the update rule in the standard REINFORCE algorithm is obtained and is given by

$$\theta_{t+1} = \theta_t + \alpha R_t \frac{\nabla \pi(a_t|s_t;\theta)}{\pi(a_t|s_t;\theta)}. \tag{5.4}$$

To enable the comparison between REINFORCE and our risk-sensitive counterpart later in Section 5.3.3, we summarize the REINFORCE algorithm in Algorithm 1.

**Algorithm 1** REINFORCE

**Input:** a differentiable policy parametrization $\pi(a|s;\theta)$.
**Algorithm parameters:** step-size $\alpha>0$, discount factor $\gamma>0$, and the risk parameter $\beta$.
**Initialization:** Initialize policy parameters $\theta \in \mathbb{R}^d$ (e.g. to 0).
**while True do**
  **Generate an episode following the policy** $\pi(\cdot|\cdot;\theta)$**, i.e.,** $s_0 \sim p_0$, $a_t \sim \pi(\cdot|s_t;\theta)$ **and** $s_{t+1} \sim p(\cdot|s_t, a_t)$**, generating a sequence of state-actions** $s_0, a_0, \ldots, s_{|\tau|-1}, a_{|\tau|-1}$
  **for** $t = 0$ **to** $|\tau| - 1$ **do**
    $\hat{R} \leftarrow \sum_{t'=t}^{|\tau|-1} \gamma^{t'-t} r_t$
    $\theta_{t+1} \leftarrow \theta_t + \alpha \gamma^t \hat{R} \nabla \log \pi(a_t|s_t;\theta)$
  **end for**
**end while**

REINFORCE is a Monte Carlo algorithm, that is, an entire trajectory needs to be generated before an update can be computed. This limits the method to episodic settings and is prone to high variance. Such high variance hinders the learning process.

#### 5.3.2.1 REINFORCE with Baseline

To further reduce the variance associated with the gradient estimations of (5.2) and (5.3), which is a must in complex environments, various techniques have been employed. Baseline methods are among the most common, and are based on subtracting an appropriately chosen baseline from the reward-to-go $R_t$ to reduce the variance without introducing bias to the estimator. Using baselines, we have [6]

$$\nabla J(\theta) \propto \mathbb{E}_{\tau\sim\rho_\theta}\Big[\sum_{t=0}^{|\tau|-1}\Big(R_t - b(s_t)\Big)\nabla \log \pi_\theta(a_t|s_t;\theta)\Big] \tag{5.5}$$

where $b(s_t)$ is a state-dependent function. State-dependent baselines are guaranteed to introduce no bias. An optimal state-dependent baseline exist, however, it is hard to find [63]. A common baseline in practice is the estimate of the value function, i.e., $b(s_t) = V^{\pi_\theta}(s_t)$, where

$$V^{\pi_\theta}(s_t) := \mathbb{E}_{\tau\sim\rho_\theta}\Big[R_t|s_t\Big]$$

Baselines have shown better convergence in practice. The effect of action-dependent baselines over state-dependent baselines is subject to debate. As we will show, a particularly convenient property of using exponential criteria is that it alleviates the need for such approaches [42]. The introduction of a baseline leads to the following algorithm. Note the change in lines 8 and 9 of Algorithm 2 compared to the REINFORCE algorithm (Algorithm 1).

**Algorithm 2** REINFORCE with Baseline (value function approximation as baseline)

**Input:** a differentiable policy parametrization $\pi(a|s;\theta)$.
**Algorithm parameters:** step-sizes $\alpha>0$ and $\bar{\alpha}>0$, discount factor $\gamma>0$, and the risk parameter $\beta$.
**Initialization:** Initialize policy parameters $\theta \in \mathbb{R}^d$ (e.g. to 0) and value parameters $w \in \mathbb{R}^{d'}$.

**while True do**
  **Generate an episode following the policy $\pi(\cdot|\cdot;\theta)$, i.e., $s_0 \sim p_0$, $a_t \sim \pi(\cdot|s_t;\theta)$ and $s_{t+1} \sim p(\cdot|s_t, a_t)$, generating a sequence of state-actions $s_0, a_0, \ldots, s_{|\tau|-1}, a_{|\tau|-1}$**
  **for $t = 0$ to $|\tau| - 1$ do**
    $\hat{R} \leftarrow \sum_{t'=t}^{|\tau|-1} \gamma^{t'-t} r_t$
    $\theta_{t+1} \leftarrow \theta_t + \alpha\gamma^t(\hat{R} - V(s_t;w_t))\nabla \log \pi(a_t|s_t;\theta)$
    $w_t = w_t + \bar{\alpha}\gamma^t(\hat{R} - V(s_t;w_t))\nabla V(s_t;w_t)$
  **end for**
**end while**

### 5.3.3 Risk-sensitive REINFORCE (R-REINFORCE)

Risk-sensitive REINFORCE (R-REINFORCE) [12] is a Monte Carlo algorithm, similar to REINFORCE [33], that seeks to find the optimal policy for the proposed risk-sensitive exponential criteria. In R-REINFORCE, the update rule is given by:

$$\nabla J_\theta(\theta) \propto \frac{1}{\beta}\mathbb{E}_{\tau\sim\rho_\theta}\Big[\sum_{t=0}^{|\tau|-1} e^{\beta R_t}\nabla \log \pi_t(\theta)\Big] \tag{5.6}$$

where $R_t := \sum_{t'=t}^{|\tau|-1} \gamma^{t'-t} r(s_{t'}, a_{t'})$. The derivation of this formula is based on a risk-sensitive variation of the policy gradient theorem [92]. These results are provided in Appendix 5.6.1. Given (5.6), the R-REINFORCE update rule reads as:

$$\theta_{t+1} = \theta_t + \frac{\alpha}{\beta}e^{\beta R_t}\frac{\nabla\pi(a_t|s_t;\theta)}{\pi(a_t|s_t;\theta)}. \tag{5.7}$$

and is a stochastic approximation algorithm (see, e.g., [93]). We provide the convergence analysis of the parameters $\theta$ in Appendix 5.6.2. The implementation of the Risk-sensitive REINFORCE algorithm is given in Alg. 3 where the differences with the (risk-neutral) REINFORCE algorithm (Alg. 1) are highlighted. For more details, the readers are referred to [12] and the references therein.

**Algorithm 3** Risk-sensitive REINFORCE

**Input:** a differentiable policy parametrization $\pi(a|s;\theta)$.
**Algorithm parameters:**
step-size $\alpha>0$, discount factor $\gamma>0$, and the risk parameter $\beta$.
**Initialization:** Initialize policy parameters $\theta \in \mathbb{R}^d$ (e.g. to 0).
**while True do**
  **Generate an episode following the policy** $\pi(\cdot|\cdot;\theta)$**,**
  **i.e.,** $s_0 \sim p_0$**,** $a_t \sim \pi(\cdot|s_t;\theta)$ **and** $s_{t+1} \sim p(\cdot|s_t, a_t)$**,**
  **generating a sequence of state-actions** $s_0, a_0, \ldots, s_{|\tau|-1}, a_{|\tau|-1}$
  **for** $t = 0$ **to** $|\tau| - 1$ **do**
    $\hat{R} \leftarrow \sum_{t'=t}^{|\tau|-1} \gamma^{t'-t} r_t$
    $\theta_{t+1} \leftarrow \theta_t + \alpha\gamma^t \frac{1}{\beta} e^{\beta\hat{R}} \nabla \log \pi(a_t|s_t;\theta_t)$
  **end for**
**end while**

Note that the update rule is not proportional to the reward-to-go $R_t := \sum_{t'=t}^{|\tau|-1} \gamma^{t'-t} r(s_{t'}, a_{t'})$, but to the exponential d

$$\beta e^{\beta R_t} = \frac{1}{\beta} \prod_{t'=t}^{|\tau|-1} \exp\{\gamma^{t'-t} \beta r(s_{t'}, a_{t'})\} \tag{5.8}$$

This is a significant difference since it yields to the multiplicative risk-sensitive Bellman's equation discussed in Section 5.4.2. In addition, notice that for the case of always positive (resp. negative) reward and negative (resp. positive) risk parameter $\beta$, a very high (resp. low) reward gets exponentially small (resp. large) weight resulting in a risk-averse behavior.

**Remark 8.** *By substituting the exponential with its Taylor series expansion (see eq. (3.3)), we can see that the risk-sensitive objective provides a natural baseline (see Section 5.3.2.1). This*

*is empirically shown in [12]. In Section 5.5, we show that such baseline leads to significant variance reduction and acceleration of learning process.*

## 5.4 Actor-Critic Methods with Exponential Criteria

Recall that REINFORCE algorithms of Alg. 1 (without baseline) and Alg. 2 (with baseline) are Monte Carlo algorithms and limited to episodic settings. To enable online update and extend to non-episodic settings, Actor-Critic algorithms [85, 94] improve the policy using gradient methods and use a critic network to estimate the value function and use it to bootstrap an estimate of the reward-to-go. That is, the estimate of the reward-to-go $R_t := \sum_{t'=t}^{|\tau|-1} \gamma^{t'-t} r(s_{t'}, a_{t'})$ is computed using the estimated/learned value function. Bootstrapping enables Actor-Critic methods to use an online update rule; eliminating the need to generate the entire trajectory before making an update and learning. This further extends the method to non-episodic settings and reduces the variance. Furthermore, Actor-critic methods [94, 95, 96] attempt to reduce the variance with replacing the Monte Carlo estimate with an estimate based on the sampled return and a function approximator. In particular, the 'actor' implements a policy gradient algorithm based on a function approximation estimated and updated by the 'critic' with every observation. Actor-critic methods have lower variance but introduce bias from the function approximation. This bias leads to instability and high sensitivity to hyperparameters. Actor-critic methods have been studied for mean-variance in [94]. Function approximation can be used to estimate $R_t$ by the value function $V^{\pi_\theta}(s_t) \simeq \mathbb{E}_{\tau\sim\rho_\theta}[R_t|s_t]$, which can be shown to satisfy the Bellman's equation

$$V^{\pi_{\theta^*}}(s) = \mathbb{E}_{a\sim\pi_{\theta^*}}\Big[r(s,a) + \gamma V^{\pi_{\theta^*}}(s')|s\Big] \tag{5.9}$$

where we use $\pi_\theta{:=}\pi(a|s;\theta)$ as a shorthand notation. The fact that (5.9) is a contraction mapping has given rise to stochastic approximation algorithms that try to asymptotically minimize the mean-squared error

$$\min_{\theta} \mathbb{E}_{a\sim\pi_\theta}\Big[\|r(s,a)+\gamma V^{\pi_\theta}(s')-V^{\pi_\theta}(s)\|^2 \mid s\Big]$$

a fact that is used by Temporal-Difference RL methods that employ learning models (e.g. neural networks [81] or other learning algorithms [97, 98]) to learn the optimal function $V^{\pi_{\theta^*}}$.

### 5.4.1 Online Actor-Critic Algorithms

Online RL methods are mainly represented by Actor-Critic (AC) methods that use two learning systems (e.g., neural networks) to estimate the parameters $\theta_t$ of the optimal policy $\pi(a_t|s_t;\theta_t)$ (actor) and the parameters $w_t$ of the value function $V(s_t;w_t)$ (critic), that is

$$\begin{cases} \theta_{t+1} = \theta_t + \alpha\Big(\hat{R}_t - V(s_t;w_t)\Big)\frac{\nabla\pi(a_t|s_t;\theta_t)}{\pi(a_t|s_t;\theta_t)} \\ w_{t+1} = w_t - \bar{\alpha}\nabla J_c(s_t;w_t,\theta_t) \end{cases} \tag{5.10}$$

where $J_c(s_t;w_t,\theta_t) = \|\hat{R}_t - V(s;w_t)\|^2$. In this case, $\hat{R}_t$ is given by

$$\hat{R}_t := r(s,\pi_{\theta_t}) + \gamma V(s';w_t)$$

where $\hat{R}_t$ is an estimate of the reward-to-go $R_t$, and the form of the objective function $J_c$ of the critic is what gives it the name Temporal-Difference (TD) learning. The algorithmic implementation is provided in Alg. 4. Special consideration needs to be given to the stepsizes $\{\alpha,\bar{\alpha}\}$ as

their choice heavily affects the learning process. The stepsizes $\{\alpha, \bar{\alpha}\}$ should decrease with time according to the theory of stochastic approximation algorithms [93, 99], i.e., $\sum_n \alpha(n) = \infty$, and $\sum_n \alpha^2(n) < \infty$, a fact that is often overlooked in practice. In addition, according to the theory of two-timescale stochastic approximation algorithms, the actor recursions should run at a higher time-scale, which is satisfied by the condition $\bar{\alpha}/\alpha \to 0$ [93].

**Algorithm 4** Online Actor-Critic

1: **Input:** a differentiable policy parametrization $\pi(a|s;\theta)$.
2: **Algorithm parameters:** step-sizes $\alpha>0$ and $\bar{\alpha}>0$, discount factor $\gamma>0$, and the risk parameter $\beta$.
3: **Initialization:** Initialize policy parameters $\theta \in \mathbb{R}^d$ and value parameters $w \in \mathbb{R}^{d'}$ (e.g. to 0) (e.g. to 0).
4: **while True do**
5: **for** $t = 0$ **to** $|\tau| - 1$ **do**
6: **Starting at an initial state** $s_0$**, take an action by following the current policy** $a_t \sim \pi(\cdot|s_t;\theta)$ **and observe the successor state** $s_{t+1} \sim p(\cdot|s_t, a_t)$**, and the reward** $r_t$
7: $\hat{R} \leftarrow r_t + \gamma V(s_{t+1}, w_t)$
8: $\theta_{t+1} \leftarrow \theta_t + \alpha\gamma^t(\hat{R} - V(s_t;w_t))\nabla \log \pi(a_t|s_t;\theta)$
9: $w_{t+1} = w_t + \bar{\alpha}\gamma^t(\hat{R} - V(s_t;w_t))\nabla V(s_t;w_t)$
10: **end for**
11: **end while**

### 5.4.2 Risk-Sensitive Bellman Equation

To introduce online updates that are not based on Monte Carlo simulation, we need to study the risk-sensitive counterparts of the Bellman equation. Recall that, in risk-neutral Reinforcement Learning, we are solving the optimal control problem

$$\max_{\pi} \mathbb{E}\left[\sum_{l=0}^{\infty} \gamma^l r(s_l, a_l)\right], \quad a_l \sim \pi(\cdot|s_l) \tag{5.11}$$

where the expectation is computed over the space of the states $s_l$ (based on the transition probabilities) while the actions are given by $a_l \sim \pi(\cdot|s_l)$. This gives rise to the definition of the

value function $V^\pi$ of a policy $\pi$ as $V^\pi(s_k) := \mathbb{E}\left[\sum_{l=k}^{\infty} \gamma^{l-k} r(s_l, a_l)|s_k\right]$. As a result of Bellman's principle, we get the (discrete-time) Hamilton-Jacobi-Bellman (HJB) equation

$$V^*(s_k) := \max_\pi \ \mathbb{E}\left[\sum_{l=k}^{\infty} \gamma^{l-k} r(s_l, a_l)|s_k\right] = \max_\pi \{\, r(s_k, a_k) + \gamma \mathbb{E}\left[V^*(s_{k+1}) \mid s_k\right] \} \tag{5.12}$$

In contrast to the risk-neutral case, in the risk-sensitive reinforcement learning setting the optimal control problem is often associated with an undiscounted version of the cost function $J_{l_\beta}$ in (3.14) and reads as:

$$\max_\pi \bar{J}_{l_\beta}(\pi) := \limsup_n \frac{1}{n} \ln \mathbb{E}\left[e^{\beta \sum_{l=0}^{n-1} r(s_l, a_l)}|s_0\right], \quad a_l \sim \pi(\cdot|s_l) \tag{5.13}$$

Notice that it has been assumed that $\gamma = 1$, and the average limit has been added to ensure boundedness of the cost. It has been shown (see, e.g., [41, 100]) that by defining a value function $\bar{V}^*_{l_\beta}(s_k) = \max_\pi \mathbb{E}\left[e^{\beta \sum_{l=k}^{t_h} r(s_l, a_l) - \ln J^*_{l_\beta}}|s_k\right]$, $a_l \sim \pi(\cdot|s_l)$, with $t_h$ being the first hitting time of a distinguished state, problem (5.13) is equivalent to a multiplicative version of the Bellman equation which defines a nonlinear eigenvalue problem:

$$\bar{V}^*_{l_\beta}(s_k) = \max_\pi \frac{e^{\beta r(s_k, a_k)}}{\bar{J}^*_{l_\beta}} \mathbb{E}\left[V^*_{l_\beta}(s_{k+1})|s_k\right], \quad a_k \sim \pi(\cdot|s_k) \tag{5.14}$$

For sufficiently small $\beta$, stochastic approximation updates in two timescales can be designed to solve the eigenvalue problem recursively implementing a policy iteration scheme and converging to an optimal stationary control that attains the optimal reward $J^*_{l_\beta} < \infty$. It is important to point out that substituting for the logarithmic value function $W(\cdot) = \ln V_{l_\beta}(\cdot)$ results in an additive dynamic programming equation, that has similarities with the classical equation for average

reward:

$$W^*(s_k) := \max_{\pi} \left\{ r(s_k, a_k) + \ln \mathbb{E}\left[ e^{W^*(s_{k+1})} \mid s_k \right] \right\} - \ln J^*_{l_\beta} \tag{5.15}$$

While this seems like a compelling formulation, and has indeed been followed by some authors (see, e.g., [37, 101]), the problem arises when attempting to formulate a reinforcement learning algorithm out of the latter dynamic programming equation. In particular, notice that, in eq. (5.15), the conditional expectation with respect to the transition probabilities appears inside a logarithm, in contrast to eq. (5.20). This typically leads to violation of the assumptions of the stochastic approximation algorithm used to train temporal-difference RL algorithms (e.g., stochastic gradient descent if using neural networks) [41]. As a result, the form of eq. (5.15) is not convenient for Q-learning and most temporal-difference RL methods. In this work, we consider the discounted optimal control problem:

$$\max_{\pi} \; J_\beta(\pi) := \frac{1}{\beta} \mathbb{E}\left[ e^{\beta \sum_{l=0}^{\infty} \gamma^l r(s_l, a_l)} \right], \quad a_l \sim \pi(\cdot|s_l) \tag{5.16}$$

as introduced in (3.22). According to the cost function $J_\beta$, we define the risk-sensitive value function of a policy $\pi$ as

$$V^\pi_\beta(s_k) := \frac{1}{\beta} \mathbb{E}\left[ e^{\beta \sum_{l=k}^{\infty} \gamma^{l-k} r(s_l, a_l)} | s_k \right], \quad a_l \sim \pi(\cdot|s_l) \tag{5.17}$$

We further define:

$$\bar{V}^\pi_\beta(s_k) := \beta V^\pi_\beta(s_k) = \mathbb{E}\left[ e^{\beta \sum_{l=k}^{\infty} \gamma^{l-k} r(s_l, a_l)} | s_k \right], \quad a_l \sim \pi(\cdot|s_l) \tag{5.18}$$

By definition, we get that $\bar{V}^{\pi}_{\beta}(\cdot) \geq 0$, and the following relationship holds:

$$\begin{aligned}
\bar{V}^{*}_{\beta}(s_k) :&= \max_{\pi} \ \mathbb{E}\left[e^{\beta \sum_{l=k}^{\infty} \gamma^{l-k} r(s_l,a_l)} | s_k\right] \\
&= \max_{\pi} \ \mathbb{E}\left[e^{\beta\left(r(s_k,a_k)+\gamma \sum_{l=k+1}^{\infty} \gamma^{l-(k+1)} r(s_l,a_l)\right)} | s_k\right] \\
&= \max_{\pi} \ e^{\beta r(s_k,a_k)} \mathbb{E}\left[(\bar{V}^{*}_{\beta})^{\gamma}(s_{k+1}) | s_k\right] \qquad (5.19) \\
&= \max_{\pi} \ \mathbb{E}\left[e^{\beta r(s_k,a_k)+\gamma \ln \bar{V}^{*}_{\beta}(s_{k+1})} | s_k\right] \qquad (5.20)
\end{aligned}$$

where $\bar{V}^{*}(\cdot) = \bar{V}^{\pi^*}(\cdot)$ is the optimal value function resulting by the optimal control policy. Notice how the use of the exponential has resulted in a multiplicative version of the Bellman equation as well. In Section 5.4.3, we will make use of this relationship to design an actor-critic approach based on the stochastic approximation updates of eq. (5.7), where a critic model will be used to recursively estimate the value function given in (5.20).

### 5.4.3 Risk-Sensitive Online Actor-Critic (R-AC)

To develop a risk-sensitive temporal-difference reinforcement learning algorithm, we use two learning systems (e.g., neural networks) to estimate the optimal policy (actor) and risk-sensitive value function (critic), similar to Section 5.4.1. That is, we keep two learning algorithms as follows:

$$\begin{cases} \theta_{t+1} = \theta_t + \alpha \frac{1}{|\beta|}\left(R^{\beta}_{t} - \bar{V}_{\beta}(s_t; w_t)\right) \frac{\nabla \pi(a_t|s_t;\theta_t)}{\pi(a_t|s_t;\theta_t)} \\ w_{t+1} = w_t - \bar{\alpha} \nabla J_r(s_t; w_t, \theta_t) \end{cases} \qquad (5.21)$$

where, in contrast to the risk-neutral case, here $R_t^\beta = \exp\left[\beta r(s_t, a_t) + \gamma \ln \bar{V}_\beta(s_{t+1}; w_t)\right]$ and $J_r$ is defined as:

$$J_r(s_t; w_t, \theta_t) = \| \exp\left[\beta r(s_t, a_t) + \gamma \ln \bar{V}_\beta(s_{t+1}; w_t)\right] - \bar{V}_\beta(s_t; w_t)\|^2, \quad a \sim \pi_{\theta_t} \tag{5.22}$$

which is a stochastic gradient descent approach to asymptotically minimize the mean-squared error:

$$\min_w \ \mathbb{E}\left[\|e^{\beta r(s_t, a_t)}(\bar{V}_\beta)^\gamma(s_{t+1}; w) - \bar{V}_\beta(s_t; w)\|^2 \mid s_t\right], \quad a \sim \pi_{\theta_t}$$

The actor parameter updates constitute a stochastic approximation algorithm based on (5.7), where the average reward-to-go $V_\beta(s_t; w_t) = \frac{1}{\beta}\mathbb{E}\left[e^{\beta R_k} | s_k\right]$ is estimated by the critic model. The critic parameter updates are again a stochastic approximation scheme that should run in a lower timescale (see Section 5.4.1) and estimate the optimal weights of the value function by minimizing the error: Notice that this recursion does not correspond to a fixed-point iteration but rather to a stochastic gradient descent approach.

**Remark 9.** *Note that simply minimizing the error* $\|\beta e^{\beta r(s,a)} + \gamma V(s'; w_t) - V(s; w_t)\|$, $a \sim \pi_{\theta_t}$, *for the value function* $V$ *defined in* (5.12) *is not equivalent to the above update rule, but it is rather equivalent to scaling the initial rewards* $r_t$ *to* $\beta e^{\beta r_t}$. *This would result in the substitution of the product term with a summation in Eq.* (5.8)*, and is a fundamentally different performance criterion.*

**Remark 10.** *The exponent* $\gamma$ *is constrained to be a rational number such that the term* $V_\beta^\gamma$ *is well-defined. However, it is not a good practice to compute the term* $V_\beta^\gamma$ *directly as it requires*

*the application of a power operation with the non-integer exponent $\gamma$. For this reason, the term $\exp(\gamma \ln V_\beta) = V_\beta^\gamma$ is used in the updates of Alg. 5, leading to a similar update law to the risk-neutral case.*

The algorithmic implementation is based on the updates in (5.21) and the objective function in (5.22) and is provided in Alg. 5. Additional comments regarding the implementation and computational difficulties of this approach are given in Remark 10. The remarks on the stepsizes $\{\alpha, \bar{\alpha}\}$ regarding the two-timescale approach hold similarly to Section 5.4.1, i.e., the iterations for the actor run in a higher timescale than these of the critic model.

**Algorithm 5** Risk-sensitive Online Actor-Critic (R-AC)

**Input:** a differentiable policy parametrization $\pi(a|s;\theta)$.
**Algorithm parameters:**
step-sizes $\alpha>0$ and $\bar{\alpha}>0$, discount factor $\gamma>0$,and the risk parameter $\beta$.
**Initialization:** Initialize policy parameters $\theta \in \mathbb{R}^d$
and value parameters $w \in \mathbb{R}^{d'}$ (e.g. to 0).
**while True do**
  **for** $t = 0$ **to** $|\tau| - 1$ **do**
    **Starting at an initial state** $s_0$,
    **take an action by following the current policy** $a_t \sim \pi(\cdot|s_t;\theta)$
    **and observe the successor state** $s_{t+1} \sim p(\cdot|s_t, a_t)$**, and the reward** $r_t$
    $\hat{R}_\beta \leftarrow \beta r_t + \gamma \ln \bar{V}_\beta(s_{t+1};w_t)$
    $\theta_{t+1} \leftarrow \theta_t + \alpha\gamma^t \frac{1}{|\beta|}(e^{\hat{R}_\beta} - \bar{V}_\beta(s_t;w_t))\nabla \log \pi(a_t|s_t;\theta)$
    $w_{t+1} \leftarrow w_t + \bar{\alpha}\gamma^t(e^{\hat{R}_\beta} - \bar{V}_\beta(s_t;w_t))\nabla \bar{V}_\beta(s_t;w_t)$
  **end for**
**end while**

## 5.5 Experiments

To evaluate the effectiveness of the proposed risk-sensitive reinforcement learning algorithms, we compare them against their risk-neutral counterparts on two classic reinforcement learning problems, namely the inverted pendulum (Cart-Pole) [102] and the underactuated dou-

ble pendulum (Acrobot) [103].

The experiments are designed to investigate the performance and robustness of the proposed risk-sensitive algorithms against model perturbations. We quantify the performance of the algorithms using the mean values of the the observed cumulative rewards $R$ during testing in different environments, and their robustness using the variance and the Conditional Value at Risk (CVaR) values, defined by:

$$\text{CVaR}_p(R) = \mathbb{E}\left[R | R \leq \text{VaR}_p(R)\right], \tag{5.23}$$

where $p$ denotes the confidence interval and the Value at Risk $\text{VaR}_p(R)$ is the $p$-quantile of the trajectory reward given by:

$$\text{VaR}_p(R) = \inf\{r \in \mathbb{R} : P(R \leq r) > p\}.$$

In particular, we make use of two $p$-quantiles for $p \in \{0.1, 0.9\}$ to capture the two tails of the distribution of $R$ (see discussion in Section 5.5.1).

### 5.5.1 On the sign and values of the risk parameter

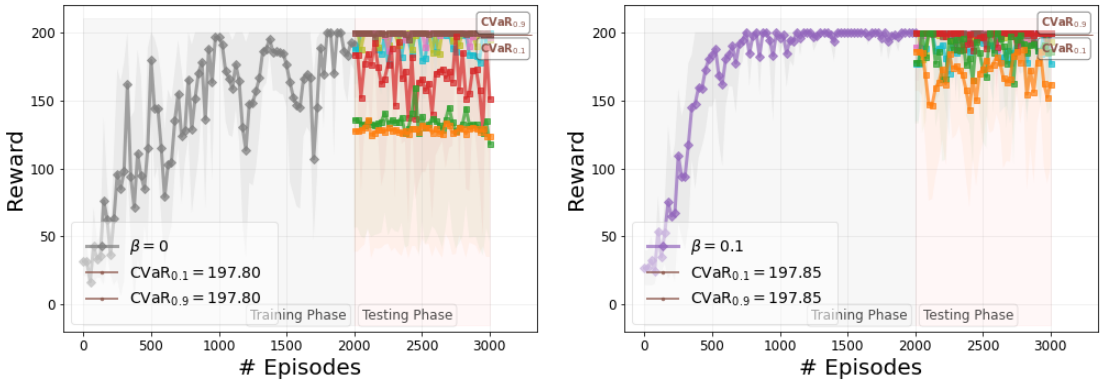
We note that in the following experiments of Sections 5.5.2 and 5.5.3, we do not optimize for the risk-sensitive hyper-parameter $\beta$. Rather, we include a comparison and a sensitivity analysis for different values of $\beta$ close to zero. As the risk-sensitive parameter $\beta$ approaches zero, the optimization of the exponential criteria approaches the risk-neutral objective. The sign of the risk parameter $\beta$ determines the optimization problem that is being solved according to (3.9).

In particular, $\beta > 0$ induces a risk-seeking (optimistic) approach, while $\beta < 0$ invokes a risk-averse approach. As shown in Section 3.4.1, the sign of the parameter $\beta$ determines which tail of the distribution of the total reward that is being weighted in the optimization process. This is consistent with our intuition that a risk-averse agent tries to optimize for the maximum average reward by weighing in the maximization of the decay of the left tail of the distribution of the total reward, while a risk-seeking agent weights in the maximization of the decay of the right tail of the reward distribution. Thus, in the simulated experiments of Sections 5.5.2 and 5.5.3, it is expected that the risk-averse approach ($\beta<0$) reduces the variance (and $\text{CVaR}_p$ values for $p > 0.5$) of the distribution of the total reward. In addition, the risk-seeking approach ($\beta>0$) does not guarantee, but can also help reduce the variance (and $\text{CVaR}_p$ values for $p < 0.5$) of the distribution of the total reward. Such a reduction can be indicative of a better suited learning behavior for the RL policies estimated by the proposed algorithm compared to the risk-neutral RL methods. Since in the risk-seeking (or "optimistic") case of $\beta>0$, emphasis is given on the right tail of the distribution of the total reward, convergence to good policies can be accelerated under certain values of the hyper-parameters of the system and certain sequences of random exploratory actions. If selecting the policies that yield the best performance among different runs (e.g. runs with different learning rates) is possible, the risk-seeking approach can also lead to better policies in terms of reduced variance. Further analysis and experimentation regarding the sensitivity of the optimization iterations with respect to the values of $\beta \in (0, \infty)$ is beyond the scope of this paper.

### 5.5.2 Inverted Pendulum (Cart-Pole)

The Cart-Pole problem is the classical inverted pendulum control problem, in which the agent is tasked to balance a pole mounted on a moving cart by an un-actuated joint [102]. The state variable of the cart-pole system has four components $(x, \theta, \dot{x}, \dot{\theta})$, where $x$ and $\dot{x}$ are the position and velocity of the cart on the track, and $\theta$ and $\dot{\theta}$ are the angle and angular velocity of the pole with the vertical. The action space consists of an impulsive "left" or "right" force $F \in \{-10, +10\}$N of fixed magnitude to the cart at discrete time intervals. A reward of $r_t = +1$ is given for each time-step $t$ that the pole kept balanced. An episode terminates successfully after $N_t = 200$ timesteps, and the average number of timesteps $\widehat{N}_t \leq N_t$ (as well as its variance) across different attempts, is used to quantify the performance of the learning algorithm. Failure occurs when $|\theta| > 12^\circ$ or when $|x| > 2.4$m.

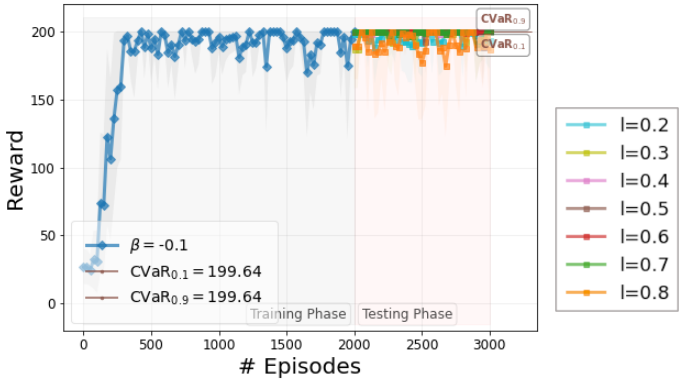
In Figure 5.1 we present the training and testing behavior of the risk-neutral REINFORCE (Alg. 1) algorithm against the proposed risk-sensitive R-REINFORCE algorithm (Alg. 3) for $\beta = -0.1$ and $\beta = +0.1$ in the Cart-Pole problem. The policy networks of the algorithms are modeled as fully connected artificial neural networks with one hidden layer of only $h = 16$ neurons and a 'ReLU' activation function. The objective functions to be optimized are as defined in Section 5.3. We use a discount factor of $\gamma = 0.99$ and the 'Adam' optimizer with the best performing learning rates within the set $\{0.001, 0.003, 0.005, 0.007, 0.01\}$ across all algorithms. The algorithms are trained for $n_e = 2000$ episodes in a training environment where the pole length is $l = 0.5$ and tested in different testing environments for $n_e = 1000$ testing runs where the length of the pole is perturbed such that $l \in [0.2, 0.8]$. The average reward for the different testing environments, as well as the $\text{CVaR}_{0.1}$, and $\text{CVaR}_{0.9}$ values for the testing environment




(a) Risk-neutral.




(b) Risk-seeking.

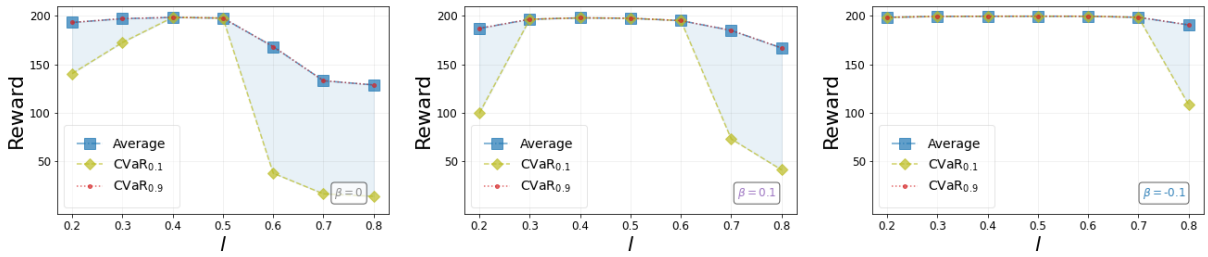



(c) Risk-averse.

Figure 5.1: Training and testing behavior of the risk-neutral REINFORCE (Alg. 1) algorithm against the proposed risk-sensitive R-REINFORCE algorithm (Alg. 3) for $\beta = -0.1$ and $\beta = +0.1$ in the Cart-Pole problem. Average reward, $\text{CVaR}_{0.1}$, and $\text{CVaR}_{0.9}$ values (for $\mathrm{l} = 0.5$) are computed over 10 independent training and testing runs with different random seeds.

without perturbations ($\mathrm{l} = 0.5$) are computed over 10 independent training and testing runs with different random seeds.

We notice that although the mean, $\text{CVaR}_{0.1}$, and $\text{CVaR}_{0.9}$ metrics are not significantly different across the three algorithms, the risk-sensitive algorithms in Fig. 5.1c and Fig. 5.1b converge faster to a near-optimal policy that shows increased robustness with respect to model perturbations. This is further assessed in Fig. 5.2, where the robustness of the algorithms with respect to model perturbations is quantified by the $\text{CVaR}_{0.1}$, and $\text{CVaR}_{0.9}$ values for all testing environments. In Fig. 5.2a, we observe that the risk-neutral REINFORCE algorithm is performing very

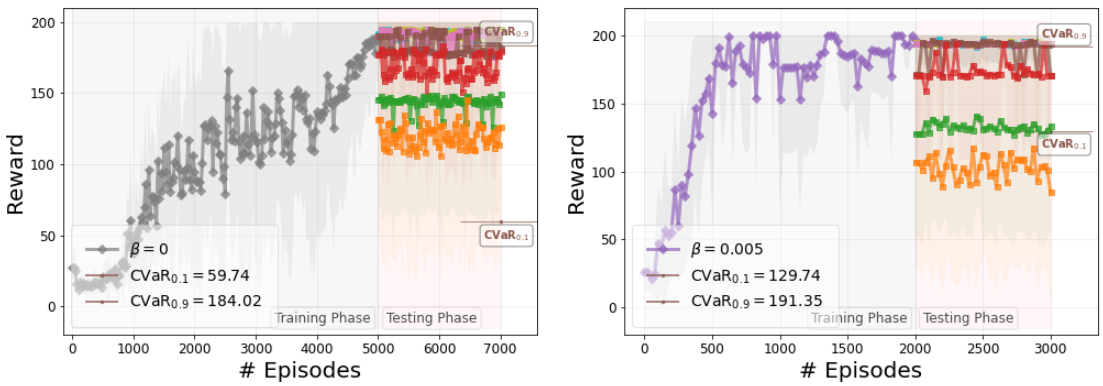



(a) $\beta = 0$. (b) $\beta = 0.1$. (c) $\beta = -0.1$.

Figure 5.2: Robustness of risk-neutral REINFORCE (Alg. 1) and risk-sensitive R-REINFORCE (Alg. 3) algorithms in a cart-pole environment with respect to varying pole length. The training environment is modeled with pole length $l = 0.5$. The testing environments have perturbed pole length values of $l \in [0.2, 0.8]$. Average reward, $\text{CVaR}_{0.1}$, and $\text{CVaR}_{0.9}$ values are computed over 10 independent training and testing runs with different random seeds.

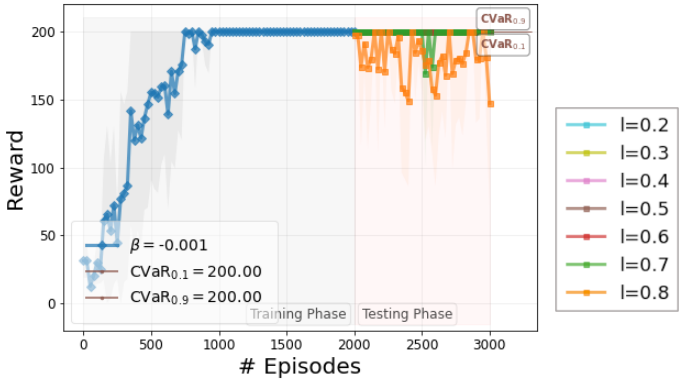
well near $l = 0.5$, i.e., where no model perturbations exist, but the performance is quickly deteriorated ($\text{CVaR}_{0.1}$ values decrease) in the presence of perturbations. Fig. 5.2b and Fig. 5.2c show that the risk-sensitive approaches increase the domain of perturbations where the behavior of the RL agent is stable, with the risk-averse approach ($\beta < 0$) showcasing the best behavior.

In Figure 5.3 we present the training and testing behavior of the risk-neutral Online Actor-Critic (OAC) (Alg. 4) and risk-sensitive actor critic (R-AC) (Alg. 5) algorithms in the cart-pole environment with respect to varying pole length. The policy networks of the algorithms are modeled as fully connected artificial neural networks with one hidden layer of only $h = 16$ neurons and a 'ReLU' activation function. The objective functions to be optimized are as defined in Section 5.4.1. We use a discount factor of $\gamma = 0.99$ and the 'Adam' optimizer with the best performing learning rates within the set $\{0.0003, 0.0005, 0.0007, 0.001\}$ across all algorithms. The algorithms are trained for $n_e = 2000$ episodes in a training environment where the pole length is $l = 0.5$ and tested in different testing environments for $n_e = 1000$ testing runs where the length of the pole is perturbed such that $l \in [0.2, 0.8]$. The average reward for the different testing environments, as well as the $\text{CVaR}_{0.1}$, and $\text{CVaR}_{0.9}$ values for the testing environment




(a) Risk-neutral.

(b) Risk-seeking.

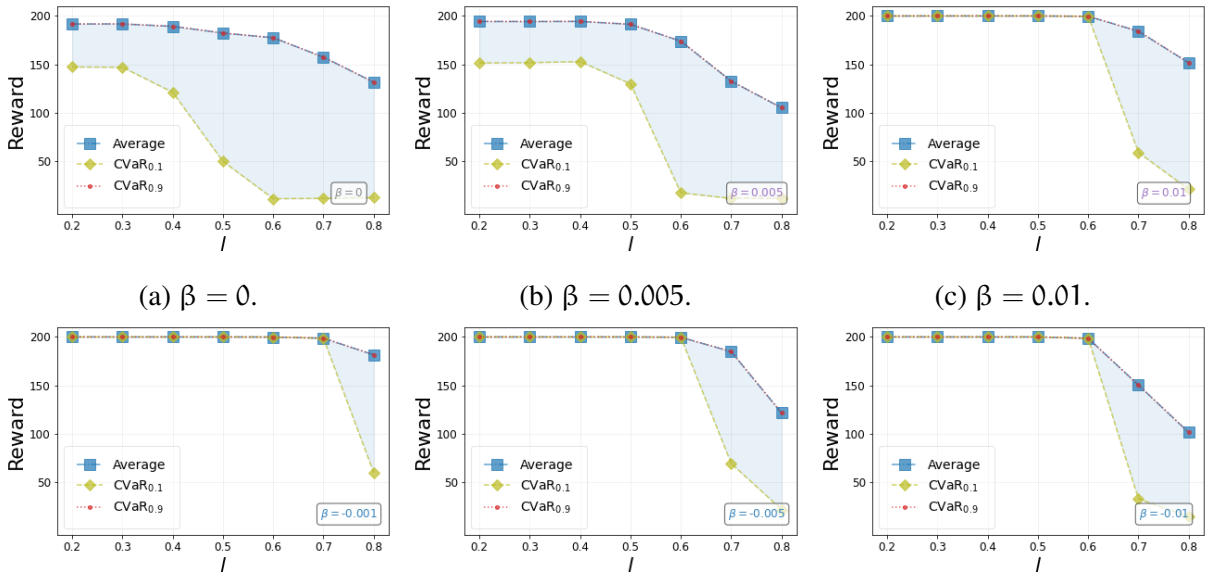



(c) Risk-averse.

Figure 5.3: Training and testing behavior of the risk-neutral Online Actor-Critic (OAC) (Alg. 4) algorithm against the proposed risk-sensitive R-AC algorithm (Alg. 5) for $\beta = -0.001$ and $\beta = +0.005$ in the Cart-Pole problem. Average reward, $\text{CVaR}_{0.1}$, and $\text{CVaR}_{0.9}$ values (for $l = 0.5$) are computed over 10 independent training and testing runs with different random seeds.

without perturbations ($l = 0.5$) are computed over 10 independent training and testing runs with different random seeds.

We notice that although the mean value performance is not significantly different across the three algorithms, the risk-sensitive algorithms in Fig. 5.3c and Fig. 5.3b converge to a near-optimal policy (in the risk-averse case the performance is optimal) that shows reduced variation across different runs, as indicated by the $\text{CVaR}_{0.1}$, and $\text{CVaR}_{0.9}$ values calculated for $l = 0.5$ (no model perturbations). Moreover, notice that the risk-neutral algorithm in 5.3a is trained for $n_e = 5000$ episodes to achieve similar performance to the risk-sensitive algorithms. This



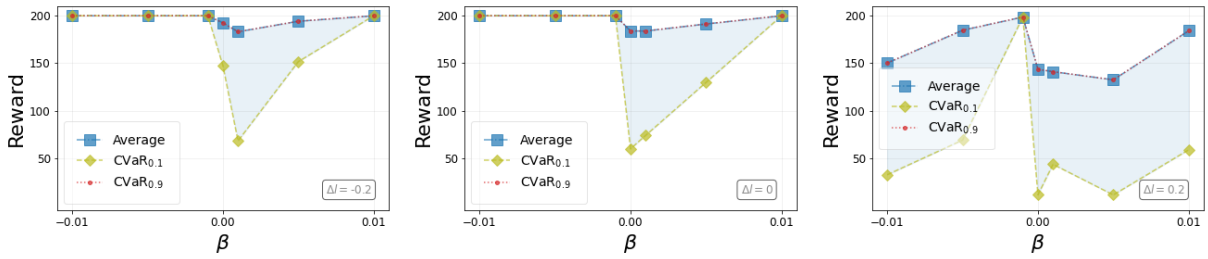

(a) $\beta = 0$. (b) $\beta = 0.005$. (c) $\beta = 0.01$.

(d) $\beta = -0.001$. (e) $\beta = -0.005$. (f) $\beta = -0.01$.

Figure 5.4: Robustness of risk-neutral Online Actor-Critic (OAC) (Alg. 4) and risk-sensitive R-AC (Alg. 5) algorithms in a cart-pole environment with respect to varying pole length. The training environment is modeled with pole length $l = 0.5$. The testing environments have perturbed pole length values of $l \in [0.2, 0.8]$. Average reward, $\text{CVaR}_{0.1}$, and $\text{CVaR}_{0.9}$ values are computed over 10 independent training and testing runs with different random seeds.

indicates better sample efficiency for the proposed risk-sensitive algorithms in Alg. 5. The robustness of the algorithms with respect to model perturbation is further assessed in Fig. 5.4. Fig. 5.4a, shows how the $\text{CVaR}_{0.1}$ values decrease as the pole length increases in the risk-neutral case ($\beta = 0$). Fig. 5.4b and Fig. 5.4c show that the risk-seeking approaches slightly increase the robustness of the learned policies. However, as shown in Fig. 5.4d, Fig. 5.4e, and Fig. 5.4f, the risk-averse approach ($\beta < 0$) showcases significantly increased robustness with respect to perturbations in the pole length.

Finally, Fig. 5.5 presents a sensitivity analysis of the algorithms with respect to the risk-sensitive parameter $\beta \in [-0.01, 0.01]$. Three testing environments are studied for $l = 0.5$ (no perturbation), $l = 0.3$ (overestimation during training), and $l = 0.7$ (underestimation during training). Negative values for $\beta$ showcase a more stable behavior across the testing environments.



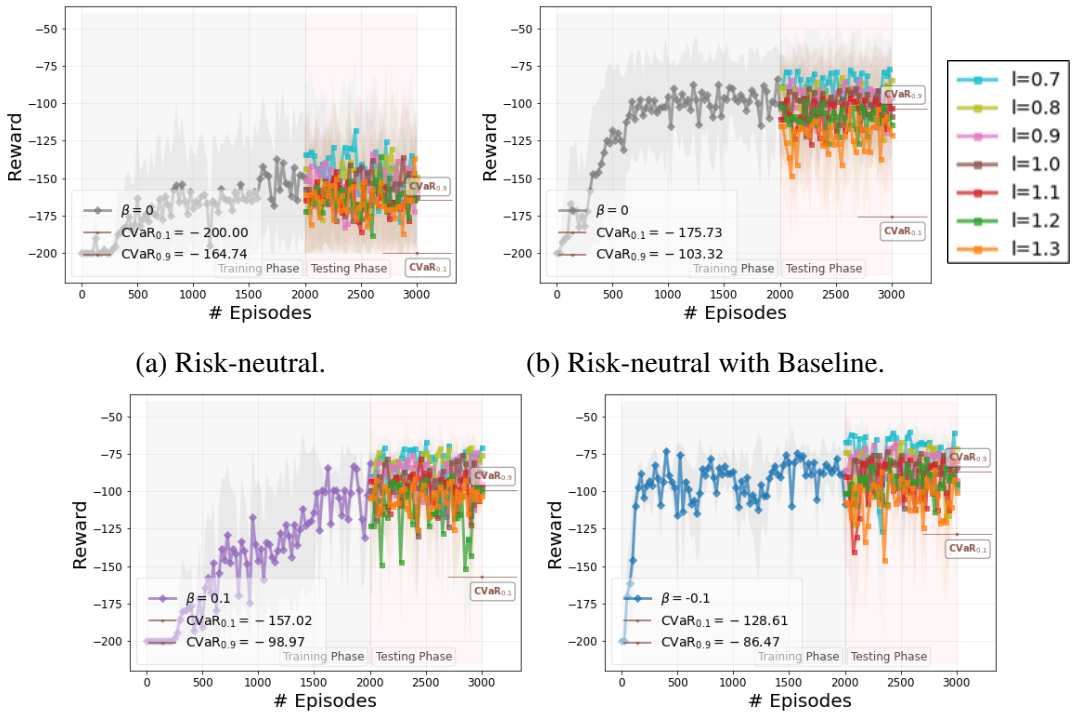

(a) $l = 0.3$. (b) $l = 0.5$. (c) $l = 0.7$.

Figure 5.5: Sensitivity analysis of the risk-sensitive R-AC algorithm (Alg. 5) with respect to the risk-sensitive parameter $\beta \in [-0.01, 0.01]$ in the Cart-Pole problem. $\beta = 0$ corresponds to the risk-neutral Online Actor-Critic (OAC) (Alg. 4). The training environment is modeled with pole length $l = 0.5$. Average reward, $\text{CVaR}_{0.1}$, and $\text{CVaR}_{0.9}$ values for testing environments with $l \in \{0.3, 0.5, 0.7\}$ are computed over 10 independent training and testing runs with different random seeds.

Moreover, notice that $\text{sgn}(\beta) < 0$ is roughly adequate for a stable behavior regardless of the numerical value of $\beta$, as long as it is close to zero, i.e., no precise estimation of the optimal $\beta$ is required.

### 5.5.3 Underactuated Double Pendulum (Acrobot)

The Acrobot problem is a double pendulum, with the joint between the two pendulum links being actuated and the other joint being un-actuated [103]. The state variable of the acrobot system has six components $(\cos\theta_1, \cos\theta_2, \sin\theta_1, \sin\theta_2, \dot{\theta}_1, \dot{\theta}_2)$, where $\theta_1$ is the angle of the first link with respect to the vertical axis (facing downwards) and $\theta_2$ is the relative angle of the second link with respect to the first link. The action space consists of a torque of $T \in \{-1, 0, +1\}$Nm of fixed magnitude applied to the actuated joint between the two links. A reward of $r_t = -1$ is given for each time-step that the double pendulum has not reached a given height. Note that the reward structure in Acrobot environment is always negative. An episode is terminated after $N_t = 200$ timesteps that the pendulum has not reached the given height.




(a) Risk-neutral.

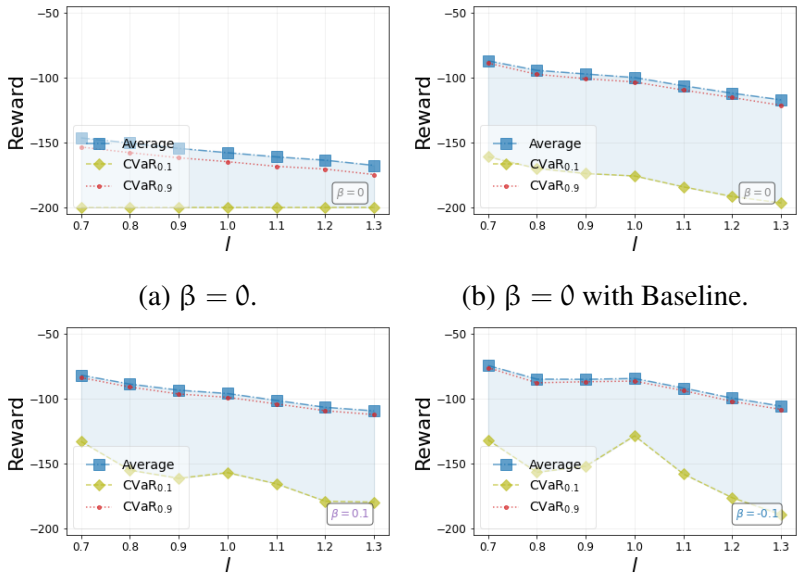
(b) Risk-neutral with Baseline.

(c) Risk-seeking.

(d) Risk-averse.

Figure 5.6: Training and testing behavior of the risk-neutral REINFORCE (Alg. 1) and risk-neutral REINFORCE with baseline (Alg. 2) algorithms against the proposed risk-sensitive R-REINFORCE algorithm (Alg. 3) for $\beta = -0.1$ and $\beta = +0.1$ in the Acrobot problem. Average reward, $\text{CVaR}_{0.1}$, and $\text{CVaR}_{0.9}$ values (for $l = 1.0$) are computed over 10 independent training and testing runs with different random seeds.

In Figure 5.6 we present the training and testing behavior of the risk-neutral REINFORCE with and without baseline (Alg. 2 and Alg. 1, respectively) algorithms against the proposed risk-sensitive R-REINFORCE algorithm (Alg. 3) for $\beta = -0.1$ and $\beta = +0.1$ in the Acrobot problem. The policy networks of the algorithms are modeled as fully connected artificial neural networks with one hidden layer of only $h = 64$ neurons and a 'ReLU' activation function. The objective functions to be optimized are as defined in Section 5.3. We use a discount factor of $\gamma = 0.99$ and the 'Adam' optimizer with the best performing learning rates within the set $\{0.001, 0.003, 0.005, 0.007, 0.01\}$ across all algorithms. The algorithms are trained for $n_e = 2000$ episodes in a training environment where the pole length of the first link is $l = 1.0$ and tested in



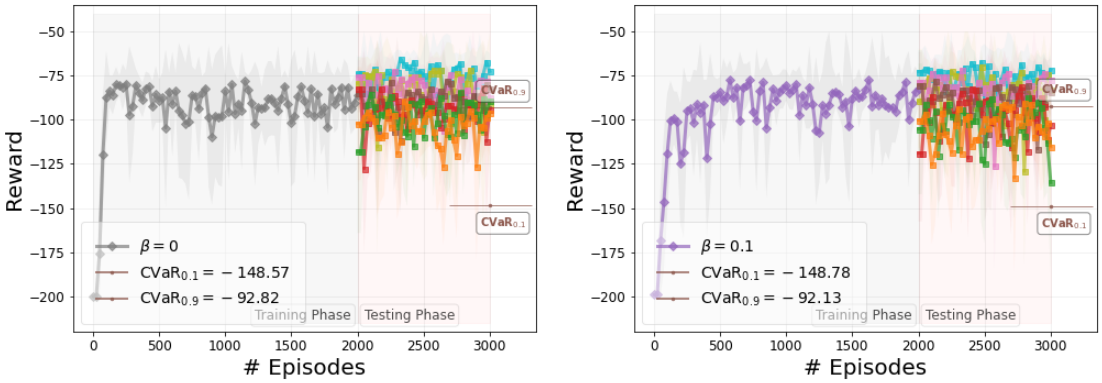

(a) $\beta = 0$. (b) $\beta = 0$ with Baseline.

(c) $\beta = 0.005$. (d) $\beta = -0.001$.

Figure 5.7: Robustness of risk-neutral REINFORCE (Alg. 1), risk-neutral REINFORCE with baseline (Alg. 2), and risk-sensitive R-REINFORCE (Alg. 3) algorithms in the Acrobot environment with respect to varying pole length. The training environment is modeled with pole length $l = 1.0$. The testing environments have perturbed pole length values of $l \in [0.7, 1.3]$. Average reward, $\text{CVaR}_{0.1}$, and $\text{CVaR}_{0.9}$ values are computed over 10 independent training and testing runs with different random seeds.

different testing environments for $n_e = 1000$ testing runs where the length of the pole is perturbed such that $l \in [0.7, 1.3]$. The average reward for the different testing environments, as well as the $\text{CVaR}_{0.1}$, and $\text{CVaR}_{0.9}$ values for the testing environment without perturbations ($l = 1.0$) are computed over 10 independent training and testing runs with different random seeds.

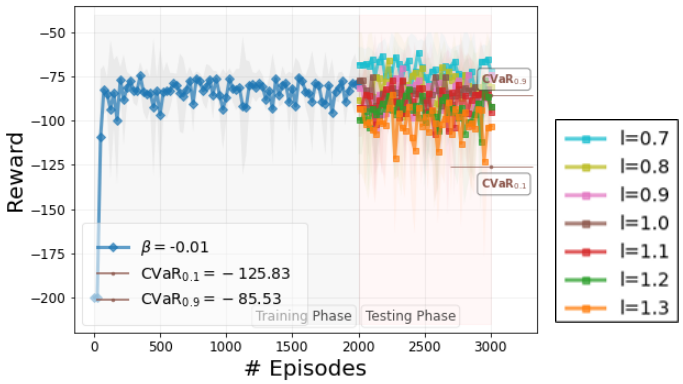
First, we notice (Fig. 5.6b) that risk-neutral REINFORCE without baseline is not able to learn a policy that solves the Acrobot problem. On the remaining algorithms, although the mean performance is not significantly different, the risk-sensitive algorithms in Fig. 5.6d and Fig. 5.6c showcase increased $\text{CVaR}_{0.1}$ values that suggest reduced variation across different runs. The fact that the risk-sensitive approaches perform on par, and slightly better, compared to REINFORCE with baseline, is indicative of the implicit baseline term present in optimizing for the exponential




(a) Risk-neutral.




(b) Risk-seeking.

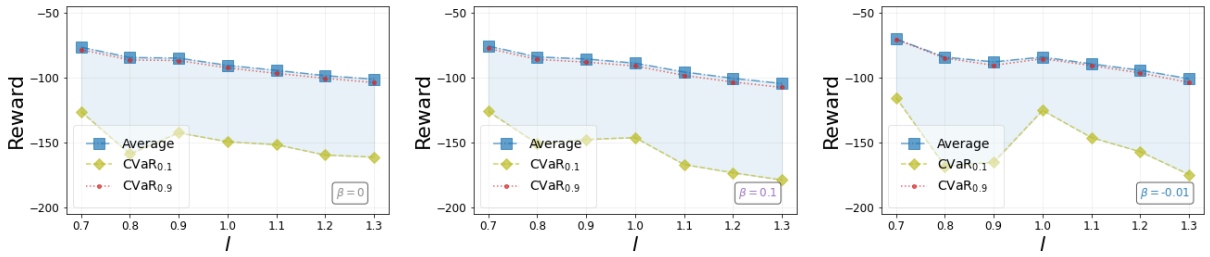



(c) Risk-averse.

Figure 5.8: Training and testing behavior of the risk-neutral Online Actor-Critic (OAC) (Alg. 4) algorithm against the proposed risk-sensitive R-AC algorithm (Alg. 5) for $\beta = -0.01$ and $\beta = +0.1$ in the Acrobot problem. Average reward, $\text{CVaR}_{0.1}$, and $\text{CVaR}_{0.9}$ values (for $l = 1.0$) are computed over 10 independent training and testing runs with different random seeds.

objective function as explained in Chapter 3. The robustness of the algorithms with respect to model perturbation is further assessed in Fig. 5.7 for all testing environments. Similar to the Cart-Pole problem, Fig. 5.7 suggests that the risk-sensitive approaches can increase the domain of perturbations where the behavior of the RL agent is more stable, with the risk-averse approach ($\beta < 0$) showcasing the best behavior.

Finally, in Figure 5.8 we present the training and testing behavior of the risk-neutral Online Actor-Critic (OAC) (Alg. 4) and risk-sensitive actor critic (R-AC) (Alg. 5) algorithms in the Acrobot environment with respect to varying pole length. The policy networks of the algorithms




(a) $\beta = 0$. (b) $\beta = 0.005$. (c) $\beta = -0.001$.

Figure 5.9: Robustness of risk-neutral Online Actor-Critic (OAC) (Alg. 4) and risk-sensitive R-AC (Alg. 5) algorithms in the Acrobot environment with respect to varying pole length. The training environment is modeled with pole length $l = 1.0$. The testing environments have perturbed pole length values of $l \in [0.7, 1.3]$. Average reward, $\text{CVaR}_{0.1}$, and $\text{CVaR}_{0.9}$ values are computed over 10 independent training and testing runs with different random seeds.

are modeled as fully connected artificial neural networks with one hidden layer of only $h = 64$ neurons and a 'ReLU' activation function. The objective functions to be optimized are as defined in Section 5.4.1. We use a discount factor of $\gamma = 0.99$ and the 'Adam' optimizer with the best performing learning rates within the set $\{0.0003, 0.0005, 0.0007, 0.001\}$ across all algorithms. The algorithms are trained for $n_e = 2000$ episodes in a training environment where the pole length of the first link is $l = 1.0$ and tested in different testing environments for $n_e = 1000$ testing runs where the length of the pole is perturbed such that $l \in [0.7, 1.3]$. The average reward for the different testing environments, as well as the $\text{CVaR}_{0.1}$, and $\text{CVaR}_{0.9}$ values for the testing environment without perturbations ($l = 1.0$) are computed over 10 independent training and testing runs with different random seeds.

Similar to the Cart-Pole case, we notice that although the mean value performance is not significantly different across the three algorithms, the risk-sensitive algorithms in Fig. 5.8c and Fig. 5.8b converge to a near-optimal policy that shows reduced variation across different runs, as indicated by the $\text{CVaR}_{0.1}$, and $\text{CVaR}_{0.9}$ values calculated for $l = 1.0$ (no model perturbations). The robustness of the algorithms with respect to model perturbation is further assessed in Fig.

5.9. Fig. 5.9a, shows how the $\text{CVaR}_{0.1}$ values decrease as the pole length increases in the risk-neutral case ($\beta = 0$). Fig. 5.9c shows that the risk-averse approach can increase the robustness of the learned policies for small perturbations.

These results are consistent with the analysis presented in Chapter 3 and suggest that the use of exponential criteria results in a risk-sensitive reinforcement learning approach that inherit computational and convergence properties of standard RL algorithms, but can also further accelerate the learning process, leading in increased sample efficiency, and result in policies with increased robustness with respect to environmental and model perturbations.

## 5.6 Appendix

### 5.6.1 Risk-Sensitive Policy Gradient Update Rule

In this section, we provide a risk-sensitive version of the policy gradient theorem [92] using exponential criteria, which is used to derive the update rule for the Risk-Sensitive REINFORCE algorithm in (5.7). Using the definition of expectation, the exponential objective can be written as an integral (summation for finite state and action spaces) over all possible trajectories, i.e.,

$$\begin{aligned}
\nabla_\theta J_\beta(\theta) &= \nabla \frac{1}{\beta} \int \rho_\theta(\tau) \exp\{\beta R(\tau)\} d\tau \\
&= \frac{1}{\beta} \int \nabla \rho_\theta(\tau) \exp\{\beta R(\tau)\} d\tau \\
&= \frac{1}{\beta} \int \rho_\theta(\tau) \frac{\nabla \rho_\theta(\tau)}{\rho_\theta(\tau)} \exp\{\beta R(\tau)\} d\tau \\
&= \frac{1}{\beta} \int \rho_\theta(\tau) \nabla \log \rho_\theta(\tau) \exp\{\beta R(\tau)\} d\tau \\
&= \frac{1}{\beta} \mathbb{E}_{\tau \sim \rho_\theta} \Big[ \nabla \log \rho_\theta(\tau) \exp\{\beta R(\tau)\} \Big]
\end{aligned} \tag{5.24}$$

Using the "log-trick" [6], the gradient of the $J_\beta(\theta)$ with respect to the policy parameter $\theta$ can be obtained as follows,

$$\begin{aligned}\nabla_\theta J_\beta(\theta) &= \nabla \frac{1}{\beta} \int \rho_\theta(\tau) \exp\{\beta R(\tau)\} d\tau \\ &= \frac{1}{\beta} \mathbb{E}_{\tau\sim\rho_\theta} \Big[ \nabla \log \rho_\theta(\tau) \exp\{\beta R(\tau)\} \Big] \end{aligned} \tag{5.25}$$

Recall that $\rho_\theta(\tau)$=$p_0 \prod_{t=0}^{|\tau|-1} \pi(a_t|s_t;\theta) p(s_{t+1}|s_t, a_t)$. Then, by first taking the logarithm and then the gradient of both sides, we get

$$\nabla \log \rho_\theta(\tau) = \sum_{t=0}^{|\tau|-1} \nabla \log \pi(a_t|s_t;\theta) \tag{5.26}$$

For brevity, we use $\pi_t(\theta)$:=$\pi(a_t|s_t;\theta)$. Thus, by substituting Eq. (5.26) in Eq. (5.24), we get

$$\nabla J_\theta(\theta) = \frac{1}{\beta} \mathbb{E}_{\tau\sim\rho_\theta} \Big[ \sum_{t=0}^{|\tau|-1} \nabla \log \pi_t(\theta) \exp\{\beta R(\tau)\} \Big] \tag{5.27}$$

Recall that $R(\tau) = \sum_{t=0}^{|\tau|-1} \gamma^t r(s_t, a_t)$. Using this fact and the property of exponential, we have

$$\nabla J_\theta(\theta) = \frac{1}{\beta} \mathbb{E}_{\tau\sim\rho_\theta} \Big[ \sum_{t=0}^{|\tau|-1} \nabla \log \pi_t(\theta) \exp\{\beta \sum_{t'=0}^{t-1} \gamma^{t'} r(s_t, a_t)\} \exp\{\beta \sum_{t'=t}^{|\tau|-1} \gamma^{t'} r(s_{t'}, a_{t'})\} \Big] \tag{5.28}$$

By using the temporal structure of the problem and causality, it can be argued that the rewards before time $t$ are not dependent on the actions that the policy will take in a future state $s_t$, that is, $\sum_{t'=0}^{t-1} \gamma^{t'} r_t(s_{t'}, a_{t'})$ is independent of $\nabla \log \pi(a_t|s_t;\theta)$. Thus, by using the independence

property, we have

$$\nabla J_\theta(\theta) = \frac{1}{\beta}\mathbb{E}_{\tau\sim\rho_\theta}\Big[\exp\{\beta\sum_{t'=0}^{t-1}\gamma^{t'}r(s_{t'},a_{t'})\}\Big]\mathbb{E}_{\tau\sim\rho_\theta}\Big[\sum_{t=0}^{|\tau|-1}\nabla\log\pi_t(\theta)\exp\{\beta\sum_{t'=t}^{|\tau|-1}\gamma^{t'}r(s_{t'},a_{t'})\}\Big] \tag{5.29}$$

Note that the first expectation is a constant, therefore,

$$\nabla J_\theta(\theta) \propto \mathbb{E}_{\tau\sim\rho_\theta}\Big[\sum_{t=0}^{|\tau|-1}\frac{1}{\beta}e^{\beta R_t}\nabla\log\pi_t(\theta)\Big] \tag{5.30}$$

where $R_t := \sum_{t'=t}^{|\tau|-1}\gamma^{t'-t}r(s_{t'},a_{t'})$.

As a final remark, notice that from (5.29), we can see that the first term on the right-hand side of the equation provides an inherent way of adjusting the step size, effectively making the constant step size adaptive.

### 5.6.2 Convergence Analysis

In this section, we show that the parameter vector $\theta$ is updated by the risk-sensitive REINFORCE algorithm in (5.7) converges to the optimal parameter vector $\theta^*$ in expectation, for sufficiently small values of the risk-parameter $\beta$. First, note the following identity

$$\begin{aligned}\|\theta_{t+1}-\theta^*\|^2 - \|\theta_t-\theta^*\|^2 &= \|\theta_{t+1}-\theta_t+\theta_t-\theta^*\|^2 - \|\theta_t-\theta^*\|^2\\ &= \|\theta_{t+1}-\theta_t\|^2 - 2(\theta_{t+1}-\theta_t)\cdot(\theta_t-\theta^*)\end{aligned}$$

Using the R-REINFORCE update rule in (5.7), i.e.,

$$\theta_{t+1} = \theta_t + \frac{\eta}{\beta} e^{\beta R_t^+} \nabla \log \pi_{\theta_t}(a_t|s_t)$$

we get

$$\begin{aligned}\|\theta_{t+1} - \theta^*\|^2 - \|\theta_t - \theta^*\|^2 = (\frac{\eta}{\beta} e^{\beta R_t^+})^2 \|\nabla \log \pi_{\theta_t}(a_t|s_t)\|^2 \\ - 2\frac{\eta}{\beta} e^{\beta R_t^+} \nabla \log \pi_{\theta_t}(a_t|s_t) \cdot (\theta_t - \theta^*)\end{aligned}$$

By taking the conditional expectation with filtration $\mathcal{F}_t$ from both sides of the equation, we have

$$\begin{aligned}\mathbb{E}\Big[\|\theta_{t+1} - \theta^*\|^2 \mid \mathcal{F}_t\Big] &= \|\theta_t - \theta^*\|^2 + (\frac{\eta}{\beta})^2 \mathbb{E}\Big[e^{2\beta R_t^+} \|\nabla \log \pi_{\theta_t}(a_t|s_t)\|^2 \mid \mathcal{F}_t\Big] \\ &\quad - 2\frac{\eta}{\beta} e^{-\beta R_t^-} \mathbb{E}\Big[e^{\beta R} \nabla \log \pi_{\theta_t}(a_t|s_t) \mid \mathcal{F}_t] \cdot (\theta_t - \theta^*) \\ &= \|\theta_t - \theta^*\|^2 + (\frac{\eta}{\beta})^2 \mathbb{E}\Big[e^{2\beta R_t^+} \|\nabla \log \pi_{\theta_t}(a_t|s_t)\|^2 \mid \mathcal{F}_t\Big] \\ &\quad - 2\eta e^{-\beta R_t^-} \nabla J_\gamma(\theta_t) \cdot (\theta_t - \theta^*)\end{aligned}$$

The first line follows from the conditioning on the filtration $\mathcal{F}_t$. The second line follows from the fact that $\nabla J_\gamma(\theta_t) = \mathbb{E}\Big[\frac{1}{\beta} e^{\beta R} \nabla \log \pi_{\theta_t}(a_t|s_t) \mid \mathcal{F}_t\Big]$. It should be noted that since $\theta^* = \mathrm{argmax}_\theta J_\gamma(\theta)$, it follows that $\nabla J_\gamma(\theta_t) \cdot (\theta_t - \theta^*) > 0$. Finally, it follows that $\theta_t$ converges to $\theta^*$, as long as the following condition holds:

$$(\frac{\eta}{\beta})^2 \mathbb{E}\Big[e^{2\beta R_t^+} \|\nabla \log \pi_{\theta_t}(a_t|s_t)\|^2 \mid \mathcal{F}_t\Big] - 2\eta e^{-\beta R_t^+} \nabla J_\gamma(\theta_t) \cdot (\theta_t - \theta^*) < 0.$$

# Chapter 6: Risk-Sensitive RL Algorithms Using PGM

## 6.1 Overview

The Probabilistic Graphical Model (PGM) framework offers systematic exploration for risk-sensitive RL. We provide a probabilistic interpretation of (I) the risk-sensitive exponential, (II) the risk-neutral expected cumulative reward, and (III) the maximum entropy Reinforcement Learning objectives, and explore their connections from a probabilistic perspective. Using Probabilistic Graphical Models (PGM), we establish that in the RL setting, maximization of the risk-sensitive exponential criteria is equivalent to maximizing the probability of taking an optimal action at all time-steps during an episode. We show that the maximization of the standard risk-neutral expected cumulative return is equivalent to maximizing a lower bound, particularly the Evidence lower Bound, on the probability of taking an optimal action at all time-steps during an episode. Furthermore, we show that the maximization of the maximum-entropy Reinforcement Learning objective is equivalent to maximizing a lower bound on the probability of taking an optimal action at all time-steps during an episode, where the lower bound corresponding to the maximum entropy objective is tighter and smoother than the lower bound corresponding to the expected cumulative return objective. These equivalences establish the benefits of risk-sensitive exponential objectives and shed light on previously postulated regularized objectives, such as maximum entropy. The utilization of a PGM model, coupled with exponential criteria, offers a

number of advantages (e.g. facilitating theoretical analysis and derivation of bounds). We then bridge the Markov Decision Process (MDP) and the PGM frameworks. We exploit the equivalence of optimizing a certain risk-sensitive criterion in the MDP formalism with optimizing a log-likelihood objective in the PGM formalism. By utilizing this equivalence, we offer an approach for developing risk-sensitive algorithms by leveraging the PGM framework. We explore the Expectation-Maximization (EM) algorithm under the PGM formalism. We show that risk-sensitive policy gradient methods can be obtained by applying sampling-based approaches to the EM algorithm, e.g., Monte-Carlo EM, with the log-likelihood. We show that Monte-Carlo EM leads to a risk-sensitive Monte-Carlo policy gradient algorithm. Our simulations illustrate the risk-sensitive nature of the resulting algorithm. The significance of these results for online learning is discussed.

## 6.2 Related Work

The Probabilistic Graphical Model (PGM) provides an alternative approach to modeling RL problems. PGMs offer a rich set of tools and techniques for inference and learning, e.g., the Expectation-Maximization (EM) algorithm, its variants, and alternatives. For a tutorial on EM algorithms, we refer the readers to [104, 105]. An elegant analysis of the EM algorithm and its convergence can be found in [106], where its fundamental nature as gradient descent in an appropriate space is established. A recent tutorial on modeling RL using PGMs is given in [107]. The use of EM-inspired algorithms in RL has been previously explored, e.g., see [108]. The survey in [109] presents some results on the EM algorithms for policy search. These results make no connection with risk-sensitive RL.

## 6.3 Modeling RL using PGM

A PGM consists of an acyclic directed graph $G=(V,E)$ and a set of properties that determine a family of probability distributions. The sets $V$ and $E$ denote the set of nodes and edges of the graph, respectively. Each node represents random variables. The edges represent conditional independence assumptions, e.g., an edge from the random variable $s_1$ to $s_2$ indicates the dependence of the random variable $s_2$ on $s_1$ (see Fig. 6.1). For nodes $a$ and $s$ in the set $V$, node $a$ is a parent of node $s$ if and only if there exists an edge from node $a$ to node $s$, e.g., $a_t$ is a parent of $s_{t+1}$ for any $t$ in Fig. 6.1. The dependence between a random variable and its parents is typically defined as a conditional distribution of the random variable represented by the node, e.g., factors of the form $p(s_{t+1}|s_t, a_t)$ for the node $s_t$ in Fig. 6.1.

PGMs can be used as an alternative framework for modeling RL problems, resulting in a PGM with factors of the form $p(s_{t+1}|s_t, a_t)$. Any feedback system can be unrolled in time and be represented by a graphical model, e.g., a Bayesian Network. This models the relationship between state $s_t$, action $a_t$ and successor-state $s_{t+1}$. Then by introducing a fictitious binary optimality variable denoted by $o_t$ at each time step, one can model the notion of reward into the graphical model; see Fig. 6.1 for a pictorial representation. Note that the system trajectory, the sequence of states $s_t$ and actions $a_t$, are observable variables and represented using white nodes. The optimality variables are unobserved variables and are represented by gray nodes. By conditioning on the optimality variables to be true, one can infer the most probable policy. An additional advantage offered by the PGM framework is the hierarchical decomposition of control and decision-making problems, "divide and conquer" approaches, and modularity, which are often necessary for the increasing complexity of systems, and problems, encountered. See

[107] for a recent tutorial on modeling RL using PGM.

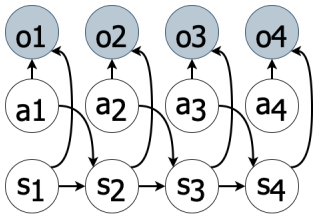



Figure 6.1: RL in the PGM Framework. The arrows represent causal dependence. The gray nodes are unobservable variables and the white nodes are observable variables.

The optimality variable is equal to one, $o_t{=}1$, if the optimal action is taken at time step $t$ and is equal to zero, $o_t{=}0$, if a non-optimal action is taken at time step $t$ (hence the name optimality variable). For brevity, we use $o_t$ and $o'_t$ to denote $o_t{=}1$ and $o_t{=}0$, respectively. We use $O_{1:T}$ to denote the event that the optimal action was taken at each time step during an episode, i.e., $O_{1:T}{=}(o_1,\cdots,o_T)$, and $O'_{1:T}$ to denote the event that the optimal action was not taken at all time steps during an episode, i.e., $O'_{1:T}{=}(o'_1,\cdots,o'_T)$. From (2.1), it follows that the joint probability of observing a trajectory and being optimal at all time steps $p_\theta(O_{1:T},\tau)$ is,

$$p_\theta(O_{1:T},\tau) = p_1 \prod_{t=1}^{T} p(a_t|s_t)p(s_{t+1}|s_t,a_t)p_\theta(o_t|s_t,a_t) \tag{6.1}$$

and the joint probability of observing a trajectory and not being optimal at all time steps is given by

$$p_\theta(O'_{1:T},\tau) = p_1 \prod_{t=1}^{T} p(a_t|s_t)p(s_{t+1}|s_t,a_t)p_\theta(o'_t|s_t,a_t), \tag{6.2}$$

where the action prior is denoted by $p(a_t|s_t)$. We assume that the action prior $p(a_t|s_t)$ is a constant corresponding to a uniform distribution over the action space. This assumption does not introduce any loss of generality, because any non-uniform prior $p(a_t|s_t)$ can be incorporated

instead into $p_\theta(o_t|s_t, a_t)$ (resp. $p_\theta(o'_t|s_t, a_t)$) via the reward function, as we shall see. The choice of the probability distribution of the optimality variable conditioned on the state-action pair $p_\theta(o_t|s_t, a_t) = p(o_t = 1|a_t, s_t)$ and $p_\theta(o'_t|s_t, a_t) = p(o_t = 0|a_t, s_t)$ defines the meaning of optimality and therefore the objective function of the agent.

## 6.4 Risk-neutral Expected Cumulative Reward Objective

In this subsection, we show that the maximization of expected cumulative reward is equivalent to maximizing a lower bound on the probability of being optimal at all time-steps during an episode. To that end, we first state three lemmas that we will use to establish our results. Lemmas 6 and 7 give the Evidence Lower Bound for $p(O_{1:T})$, and multiple approaches to prove them have been suggested in the literature. For the convenience of the reader and completeness, we include one approach for showing the bound here [110]. The proof for lemma 8 is of our own. Then, we formally state our results in Theorem 3 with the proof immediately following the theorem. We end this section with a brief discussion.

**Lemma 6.** *The following equality holds:*

$$\log p(O_{1:T}) = \mathbb{E}_{\tau\sim\rho_\theta}\Big[\log p(O_{1:T}|\tau)\Big] + D\Big(\rho_\theta(\tau)\|p(\tau|O_{1:T})\Big)$$

*where* $D(Q, P)$ *is the KL-divergence between the probability distributions* $Q$ *and* $P$.

*Proof of Lemma 6.*

$$
\begin{aligned}
D(\rho_\theta(\tau)\|p(\tau|O_{1:T})) &= \mathbb{E}_{\tau\sim\rho_\theta}\Big[\log\frac{\rho_\theta(\tau)}{p(\tau|O_{1:T})}\Big] \\
&= -\mathbb{E}_{\tau\sim\rho_\theta}\Big[\log\frac{p(\tau|O_{1:T})}{\rho_\theta(\tau)}\Big] \\
&= -\mathbb{E}_{\tau\sim\rho_\theta}\Big[\log\frac{p(\tau,O_{1:T})}{\rho_\theta(\tau)} - \log p(O_{1:T})\Big] \\
&= -\mathbb{E}_{\tau\sim\rho_\theta}\Big[\log\frac{p(\tau,O_{1:T})}{\rho_\theta(\tau)}\Big] + \log p(O_{1:T}) \\
&= -\mathbb{E}_{\tau\sim\rho_\theta}\Big[\log p(O_{1:T}|\tau)\Big] + \log p(O_{1:T})
\end{aligned}
$$

The first line follows from the definition of Kullback-Leibler divergence. The second and third lines follow form properties of the logarithm, and the forth line follows straightforward from the fact that $p(O_{1:T})$ is a constant with respect to $\tau$. The last line follows from the definition of conditional probability. Thus, by rearranging the terms, one can obtain equality. □

**Lemma 7.** *The following lower bound holds on the probability of taking an optimal action at all time-steps during an episode* $p(O_{1:T})$*:*

$$\log p(O_{1:T}) \geq \mathbb{E}_{\tau\sim\rho_\theta}\Big[\log p(O_{1:T}|\tau)\Big]$$

*Proof of Lemma 7.* It follows straightforward from lemma 6 and non-negativity of KL-divergence. □

The lower bound presented in lemma 7 is the Evidence Lower Bound on the probability of

taking an optimal action at all time-steps during an episode, i.e.,

$$L = \mathbb{E}_{\tau\sim\rho_\theta}\Big[\log p(O_{1:T}|\tau)\Big] \tag{6.3}$$

**Lemma 8.** *The Evidence Lower bound can be expressed as:*

$$L = \beta\mathbb{E}_{\tau\sim\rho_\theta}\Big[R(\tau)\Big] + \sum_{t=1}^{T}\log p(a_t|s_t) \tag{6.4}$$

*Proof of lemma 8.* It follows straightforward from lemma 7 that the Evidence lower bound is

$$\begin{aligned} L &= \mathbb{E}_{\tau\sim\rho_\theta}\Big[\log p(O_{1:T},\tau)\Big] - \mathbb{E}_{\tau\sim\rho_\theta}\Big[\log \rho_\theta(\tau)\Big] \\ &= \mathbb{E}_{\tau\sim\rho_\theta}\Big[\beta R(\tau)\Big] + \mathbb{E}_{\tau\sim\rho_\theta}\Big[\sum_{t=1}^{T}\log p(a_t|s_t)\Big] \\ &= \beta\mathbb{E}_{\tau\sim\rho_\theta}\Big[R(\tau)\Big] + \sum_{t=1}^{T}\log p(a_t|s_t) \end{aligned}$$

The first line follows from definition of conditional probability. The second line is obtained by substituting Eq's (2.1) and (6.1), substituting of $p(o_t|s_t,a_t)=\pi(a_t|s_t;\theta)e^{\beta r_t}$ in Eq. (6.1), and then using the sum property of logarithm. □

Now, we are ready to state and proof our theorem.

**Theorem 3.** *The maximization of expected cumulative reward objective, i.e.,*

$$J(\theta) := \mathbb{E}_{\tau\sim\rho_\theta}\Big[R(\tau)\Big]$$

*is equivalent to maximization of the Evidence Lower Bound on the probability of being optimal*

*at all time-steps during an episode, i.e.,* $L = \mathbb{E}_{\tau \sim \rho_\theta}\Big[\log p(O_{1:T}|\tau)\Big]$.

*Proof.* It follows straightforward from lemmas 7 and 8 and positivity of $\beta$ that

$$L \propto \mathbb{E}_{\tau \sim \rho_\theta}\Big[R(\tau)\Big] + (1/\beta)\sum_{t=1}^{T} \log p(a_t|s_t) \tag{6.5}$$

By noting that the action prior $p(a_t|s_t)$ is a uniform distribution over action space, and consequently, $\sum_{t=1}^{T} p(a_t|s_t)$ is a constant and does not depend on $\theta$, we can see that the optimization of the Evidence Lower Bound is equivalent to maximizing the expected cumulative reward. □

Theorem 3 can be interpreted to state that the expected cumulative reward objective attempts to approximately optimize the probability of being optimal at all time steps during an episode by maximizing a lower bound, particularly the Evidence Lower Bound, on the probability of being optimal at all time steps during an episode.

## 6.5 Maximum Entropy Objective

We explore the connections of the maximum entropy objective with exponential criteria and expected cumulative return objective, and offer a mathematical and intuitive explanation for the maximum entropy objective, which justifies why maximum entropy objective results in more robust and improved policies. In particular, we show that the maximization of maximum entropy objective is an attempt to approximately solve a multi-objective optimization using scalarization method, which in turn is equivalent to maximization of a tighter and smoother lower bound on the probability of taking an optimal action at all time-steps during an episode than the Evidence Lower Bound corresponding to the maximization of the risk-neutral objective.

To that end, we first prove a lemma that in conjunction with our previous lemmas and theorems will help to establish the maximum entropy's connection to exponential and expected cumulative return objectives. We end this section by summarizing the connection between these objectives.

**Lemma 9.** *The KL-divergence term in lemma 6 can be expressed in terms of the policy's entropy as follows*

$$\begin{aligned} KL &= D\Big(\rho_\theta(\tau) \| p(\tau|O_{1:T})\Big) \\ &= -\mathbb{E}_{s_{t+1}\sim p(s_{t+1}|s_t,a_t)}\Big[\sum_{t=1}^{T} \mathcal{H}^{\pi}(\cdot|s_t)\Big] - \sum_{t=1}^{T} \log p(a_t|s_t) \end{aligned}$$

*proof of Lemma 9.*

$$\begin{aligned} KL &= D\Big(\rho_\theta(\tau) \| p(\tau|O_{1:T})\Big) \\ &= \mathbb{E}_{\tau\sim\rho_\theta}\Big[\log \frac{\rho_\theta(\tau)}{p(\tau|O_{1:T})}\Big] \\ &= \mathbb{E}_{\tau\sim\rho_\theta}\Big[\log \frac{\rho_\theta(\tau) p(O_{1:T})}{p(O_{1:T},\tau)}\Big] \\ &= \mathbb{E}_{\tau\sim\rho_\theta}\Big[\log \frac{\prod_t \pi(a_t|s_t;\theta)}{\prod_t p(a_t|s_t)}\Big] \\ &= \sum_{t=1}^{T} \mathbb{E}_{\tau\sim\rho_\theta}\Big[\log \pi(a_t|s_t;\theta)\Big] - \sum_{t=1}^{T} \log p(a_t|s_t) \end{aligned}$$

The first line follows from the definition of Kullback–Leibler (KL) divergence. The second line follows for the fact that for any two random variables X and Y, $p(X|Y)$=$p(X,Y)p(X)/p(Y)$. The third equality follows from substituting $\rho_\theta(\tau)$ and $p(O_{1:T},\tau)$ using Eq's (2.1) and (6.1), and noting that the $p(O_{1:T}) = \prod_{t=1}^{T} p(o_t|a_t,s_t)$.

By noting that $\mathbb{E}_{\tau\sim\rho_\theta}[\cdot]$ is an equivalent notation for $\mathbb{E}_{s_t\sim p(s_{t+1}|s_t;a_t)}\mathbb{E}_{a_t\sim\pi(a_t|s_t;\theta)}[\cdot]$, we have

$$\begin{aligned}
KL &= \sum_{t=1}^{T} \mathbb{E}_{s_t\sim p(s_{t+1}|s_t;a_t)}\Big[\mathbb{E}_{a_t\sim\pi(a_t|s_t;\theta)}\Big[\log\pi(a_t|s_t;\theta)\Big]\Big] \\
&\quad - \sum_{t=1}^{T}\log p(a_t|s_t) \\
&= -\mathbb{E}_{s_t\sim p(s_{t+1}|s_t;a_t)}\Big[\sum_{t=1}^{T}\mathcal{H}^{\pi}(\cdot|s_t)\Big] - \sum_{t=1}^{T}\log p(a_t|s_t)
\end{aligned}$$

where $\mathcal{H}^{\pi}(\cdot|s_t)=-\mathbb{E}_{a_t\sim\pi(a_t|s_t;\theta)}\Big[\log\pi(a_t|s_t;\theta)\Big]$ is the definition of shannon entropy. □

Now, we are ready to explore the maximum entropy connections to the exponential and expected cumulative return objectives. Using lemma 6, we can see that the gap between the logarithm of probability of taking an optimal action at all time steps during an episode $\log p(O_{1:T})$, which is the objective that the exponential criteria optimize, and the Evidence lower bound $L$ (cf. Eq. (6.3)), which is the objective that expected cumulative return optimizes, is

$$G = D\Big(\rho_\theta(\tau)\|p(\tau|O_{1:T})\Big)$$

**Theorem 4.** *Under the assumption of negative reward structure, the maximization of the maximum entropy objective of Eq.* (4.8)*,i.e.,*

$$J_{ent}(\theta) := \mathbb{E}_{\tau\sim\rho_\theta}\Big[R(\tau)\Big] + \lambda\mathbb{E}_{\substack{s_1\sim p_1\\ s_t\sim p(\cdot|s_{t-1},a_{t-1})}}\Big[\sum_{t=1}^{T}\mathcal{H}^{\pi}(a_t|s_t)\Big]$$

*is equivalent to maximizing a linear scalarization, i.e. a weighted sum of the objective functions,*

*of the following multi-objective optimization*

$$\max_{\theta} \quad (L, -G)$$

*where* $L = \mathbb{E}_{\tau \sim \rho_\theta}\Big[\log p(O_{1:T}|\tau)\Big]$ *(cf. Eq. (6.3)) is the Evidence Lower Bound on the probability of taking an optimal action at all time-steps during an episode, and* $G = D\Big(\rho_\theta(\tau) \| p(\tau|O_{1:T})\Big)$ *(cf. Eq. (6.6)) is the gap between the Evidence Lower Bound and the log probability.*

*Proof of Theorem 4.* Using lemma 9, we have

$$G = -\mathbb{E}_{s_t \sim p(s_{t+1}|s_t, a_t)}\Big[\sum_{t=1}^{T} \mathcal{H}^{\pi}(\cdot|s_t)\Big] - \sum_{t=1}^{T} \log p(a_t|s_t) \tag{6.6}$$

Also, recall that using lemma 8, we have the Evidence Lower Bound $L$.

The proof is complete by scalarizing the multi-objective optimization as a single-objective optimization with a scalarization weight corresponding to the regularization weight $\lambda\beta$, and noting that the action prior $p(a_t|s_t)$ is a uniform distribution and therefore is constant with respect to $\theta$. □

Theorem 4 shows that the maximum entropy objective is equivalent to a linear scalarization of the multi-objective optimization involving simultaneously maximizing the Evidence Lower Bound on the probability of taking an optimal action at all time-steps during an episode and minimizing the gap between the log probability and the Evidence Lower Bound.

Thus, the maximization of maximum entropy objective can be thought of as an attempt to trade-off the tightness of the lower bound on the log probability of taking an optimal action at all time-steps during an episode and the optimization of the lower bound, effectively trying

to find and optimize a smooth lower bound, tighter than the Evidence Lower Bound, on the log probability $\log p(O_{1:T})$.

The solution to the maximum entropy objective is a Pareto optimal solution of the multi-objective problem corresponding to the given regularization weight $\lambda$. Given this perspective, one can use alternative methods to solve the multi-objective optimization, which might lead to more effective algorithms.

## 6.6 Risk-Sensitive RL

In this section, we briefly introduce the risk-sensitive exponential criteria (under the MDP framework) and state its connection with a certain log-likelihood objective under the PGM framework. This connection enables the development of risk-sensitive RL algorithms under the PGM framework. Recall that the exponential criterion is a particular example of such criteria and is given by

$$J_\beta(\theta) := \mathbb{E}\left[\beta e^{\beta R}\right]$$

where the expectation is taken under policy's trajectory distribution and $\beta \in \mathbb{R}$ is a real-value constant design parameter that controls the agent's risk-attitude. The agent shows risk-averse behavior for a negative risk-parameter $\beta < 0$ and risk-seeking behavior for a positive risk-parameter $\beta > 0$ [43].

The results in [48] give a probabilistic interpretation of maximizing the exponential criterion (3.1) by casting the risk-sensitive RL problem into the PGM framework, and establish that, under a bounded reward model $r \in [r_{min}, r_{max}]$, the maximization of the exponential criterion is

equivalent to maximizing the probability of taking an optimal action at all time steps during an episode for a factored form of

$$p_\theta(o_t \mid s_t, a_t) := \pi(a_t \mid s_t; \theta) e^{\beta r_t}.$$

That is, the specific choice of $p_\theta(o_t \mid s_t, a_t) := \pi(a_t|s_t;\theta)e^{\beta r_t}$ maps back to a specific objective function in the MDP framework, namely, the exponential criteria with a positive risk parameter $\beta > 0$ (risk-seeking behavior). [48] does not leverage this equivalence and offers no algorithm. To leverage this equivalence and design a risk-sensitive RL algorithm, we first extend the results in [48] to characterize both positive (risk-seeking behavior) and negative (risk-averse behavior) risk parameters and formally state this connection in Theorem 10. We then use this equivalence to leverage EM algorithms for developing risk-sensitive RL in the proceeding section.

**Theorem 10.** *Under the assumption of bounded reward structure, for the choice of* $p(o_t|s_t, a_t) = \pi_\theta(a_t|s_t)e^{\beta r_t}$ *with the temperature parameter* $1/\beta > 0$*, we have*

$$\arg\max_\theta J_\beta(\theta) = \arg\max_\theta \log p_\theta(O_{1:T}), \quad \forall \beta > 0$$

*and for the choice of* $p(o'_t|s_t, a_t) = \pi_\theta(a_t|s_t)e^{\beta r_t}$ *with the temperature parameter* $1/\beta < 0$*, we have*

$$\arg\max_\theta J_\beta(\theta) = \arg\min_\theta \log p_\theta(O'_{1:T}), \quad \forall \beta < 0.$$

*Proof.* The proof is presented in the appendix. □

**Remark 11.** *Theorem 10 suggests that maximizing the joint probability of taking an optimal*

*action at all time steps during an episode, i.e.,*

$$\arg\max_{\theta} \log p_\theta(O_{1:T}), \tag{6.7}$$

*results in a risk-seeking behavior, and minimizing the joint probability of not taking an optimal action at all time steps during an episode, i.e.,*

$$\arg\min_{\theta} \log p_\theta(O'_{1:T}), \tag{6.8}$$

*results in risk-averse behavior. This also provides a probabilistic view of risk-sensitive RL with exponential criterion and justifies the choice of this criterion as a reasonable objective for an RL agent.*

The equivalence stated in Theorem 10 suggests an avenue for developing risk-sensitive RL algorithms by leveraging numerous existing methods for optimizing log probabilities. A wealth of tools and algorithms for inference and learning on graphical models, e.g. the EM algorithm, have been developed over the years [104, 105]. These tools can be used to (approximately) solve the risk-sensitive RL problem. The objective functions in (6.7) and (6.8) are derived from the mathematical analysis of the risk-sensitive exponential criterion under the PGM framework and are different from the objective functions used in methods described in [109].

Theorems 10 and 3 show that the optimization of the risk-sensitive exponential criteria is equivalent to maximizing the joint probability of taking and optimal action at all time-steps during an episode, resulting in a risk-seeking behavior, while the expected cumulative reward is an attempt to approximately solve this optimization by optimizing a lower bound on the probability

of taking an optimal action at all time-steps.

## 6.7 Monte-Carlo EM Algorithms and Risk-Sensitive RL

We focus on the EM algorithm and its sampling-based variant, Monte-Carlo EM. We will explore other EM variants and alternatives for developing risk-sensitive RL algorithms in our future work. We formally state our result in Theorem 5. The proof of Theorem 5 is presented in sections 6.8.

**Theorem 5.** *Let* $R_t := \sum_{t'=t}^{T-1} \gamma^{t'-t} r_{t'}$ *denote the reward-to-go at time* $t$ *and the parameter* $\alpha$ *to be the step size. Then, the iteration*

$$\theta^{k+1} = \theta^k + \alpha e^{\beta R_t} \nabla_\theta \log \pi_\theta(a_t|s_t) \tag{6.9}$$

*is a Monte-Carlo EM to solve the risk-sensitive exponential criteria* (3.1).

*Proof.* The proof uses the sampling-based Monte-Carlo EM and gradient descent algorithms. See Section 6.8 for a step-by-step proof. □

**Remark 12.** *The update rule in Theorem 5 is the risk-sensitive policy gradient algorithm proposed in [12] that has been derived using the MDP framework. This suggests that risk-sensitive policy gradient algorithms can be thought of as sample- and gradient-based EM algorithms.*

## 6.8 Proof of Theorem 5

Here, we provide a step-by-step proof of the theorem. The proof is the application of the EM algorithm to the RL problem under the PGM framework. We first state a lemma. The lemma

is later used to construct a lower bound on the posterior distribution for the EM algorithm. We write the proof for the risk-seeking $\beta > 0$ case (the optimization of (6.7)). The proof for the risk-averse $\beta < 0$ (the optimization of (6.8)) case is analogous by substitution of $O'_{1:T}$ for $O_{1:T}$ and noting the negativity of $\beta$.

**Lemma 11.** *[48] The log-likelihood in Eq. (6.7) can be decomposed as the sum of the Evidence lower-bound on the probability of being optimal at all time steps and a KL-divergence term, given by*

$$\log p_\theta(O_{1:T}) = L(\theta) + D(\theta)$$

*where*

$$L(\theta) := \mathbb{E}\Big[\log p(O_{1:T}|\tau)\Big]$$

$$D(\theta) := D_{KL}\Big(p_\theta(\tau)\|p_\theta(\tau|O_{1:T})\Big)$$

$D_{KL}(Q, P)$ *is the KL-divergence between the probability distributions* $Q$ *and* $P$.

**Remark 13.** *It should be noted that the optimality of a trajectory is not dependent on the policy, that is to say,* $p(O_{1:T}|\tau)$ *is independent of the policy, and thus, does not depend on the policy parameters* $\theta$.

**Remark 14.** *It is immediate from Lemma 6 and the non-negativity of KL divergence that* $L(\theta)$ *is a lower bound on the log probability.*

We now present a step-by-step derivation and discussion of the theorem. Recall that in the EM algorithm, the E-step constructs a tractable lower bound $B(\theta, \theta_t)$ and the M-step maximizes the constructed bound. The E-step constructs a posterior distribution lower bound, see Lemma 11. The E-step updates the trajectory distribution $p_\theta(\tau)$ by minimizing the KL divergence term $D_{KL}(p_\theta(\tau) \| p_{\theta^k}(\tau|O_{1:T}))$, i.e.,

$$\theta_e^* := \arg\min_\theta D_{KL}\Big(p_\theta(\tau) \| p_{\theta^k}(\tau|O_{1:T})\Big).$$

Note that the KL divergence is non-negative and is minimized when the parameters are chosen such that $p_{\theta_e^*}(\tau) = p_{\theta^k}(\tau|O_{1:T})$. This lower bound is tight after each E-step. Thus, the M-step maximizes the constructed lower bound, i.e.,

$$\theta^{k+1} = \arg\max_\theta B(\theta, \theta^k)$$

where $B$ is a lower bound and is given by

$$B(\theta, \theta^k) := \mathbb{E}_{p_{\theta^k}(\tau|O_{1:T})}\Big[\log p_\theta(O_{1:T}, \tau)\Big].$$

By noting that $p(O_{1:T}|\tau)$ is independent of the policy parameters $\theta$, we have

$$p_\theta(O_{1:T}, \tau) = p(O_{1:T}|\tau) p_\theta(\tau)$$

and by Bayes' rule, we have

$$p_{\theta^k}(\tau|O_{1:T}) = \frac{p(O_{1:T}|\tau)p_{\theta^k}(\tau)}{p_{\theta^k}(O_{1:T})}.$$

Thus,

$$\begin{aligned} B(\theta, \theta^k) &= \mathbb{E}_{\tau\sim p_{\theta^k}(\cdot)}\Big[\frac{p(O_{1:T}|\tau)}{p_{\theta^k}(O_{1:T})}\log p_\theta(O_{1:T},\tau)\Big] \\ &= \mathbb{E}_{\tau\sim p_{\theta^k}(\cdot)}\Big[\frac{p(O_{1:T}|\tau)}{p_{\theta^k}(O_{1:T})}\log p_\theta(\tau)\Big] \\ &\qquad + \mathbb{E}_{\tau\sim p_{\theta^k}(\cdot)}\Big[\frac{p(O_{1:T}|\tau)}{p_{\theta^k}(O_{1:T})}\log p(O_{1:T}|\tau)\Big]. \end{aligned}$$

The second term is not a function of the decision variable $\theta$, thus, we have

$$\theta^{k+1} = \arg\max_\theta \mathbb{E}_{\tau\sim p_{\theta^k}(\cdot)}\Big[\frac{p(O_{1:T}|\tau)}{p_{\theta^k}(O_{1:T})}\log p_\theta(\tau)\Big].$$

By noting $p_{\theta^k}(O_{1:T})$ is constant with respect to $\tau$ and not a function of $\theta$ (it is a function of $\theta^k$), to compute the new parameter, we have

$$\theta^{k+1} = \arg\max_\theta \mathbb{E}_{\tau\sim p_{\theta^k}(\cdot)}\Big[p(O_{1:T}|\tau)\log p_\theta(\tau)\Big].$$

Now recall that

$$p(O_{1:T}|\tau) = \prod_{t=1}^{T} p(o_t|s_t, a_t) = \prod_{t=1}^{T} e^{\beta r_t} = e^{\beta R}.$$

Thus,

$$\theta^{k+1} = \arg\max_{\theta} \mathbb{E}_{\tau\sim p_{\theta^k}(\cdot)}\left[e^{\beta R}\log p_\theta(\tau)\right].$$

By noting that $\log p_\theta(\tau)=\sum_{t=1}^{T}\log \pi_\theta(a_t|s_t)$, we arrive at the following update rule:

$$\theta^{k+1} = \arg\max_{\theta} \mathbb{E}_{\tau\sim p_{\theta^k}(\cdot)}\left[e^{\beta R}\sum_{t=1}^{T}\log \pi_\theta(a_t|s_t)\right]. \tag{6.10}$$

**Remark 15.** *The update rule (6.10) has been used in [111] and [109], but the connection with risk-sensitive RL has not been made.*

**Remark 16** (MM Principle)**.** *It is well known that EM algorithms can be thought of as an instance of Majorization-Minimization (MM) algorithms. The bound* $B(\theta,\theta^k)$ *minorizes the log-likelihood* $\log p_\theta(O_{1:T})$*, that is, it is tangent to the log-likelihood at a given policy parameter* $\theta^k$ *and it is dominated by it at all points, i.e.,*

$$B(\theta^k,\theta^k) = \log p_{\theta^k}(O_{1:T})$$

$$B(\theta,\theta^k) \leq \log p_\theta(O_{1:T}) \qquad \forall\theta\in\mathbb{R}^d.$$

*This suggests one could use alternative MM-type algorithms to develop risk-sensitive RL agents.*

The iteration (6.10) is an EM algorithm to solve the risk-sensitive exponential criteria (3.1). To adapt the update rule of (6.10) to use the interactions with the environment, one can use sampling-based EM algorithms, such as Monte-Carlo EM. Given the iteration of (6.10), by the

linearity of expectation, the resulting EM iteration of Eq (6.10) can be expressed as

$$\theta^{k+1} = \arg\max_{\theta} \sum_{t=1}^{T} \mathbb{E}_{\tau \sim p_{\theta^k}(\cdot)} \left[ e^{\beta R} \log \pi_\theta(a_t|s_t) \right].$$

The optimization in Eq (6.10) may be attempted by a variety of methods. One particularly easy-to-implement approach is using a first-order (gradient-based) method and leads to the following update rule

$$\theta_{l+1}^{k+1} = \theta_l^{k+1} + \alpha \sum_{t=1}^{T} \mathbb{E}_{\tau \sim p_{\theta^k}(\cdot)} \left[ e^{\beta R} \nabla_\theta \log \pi_\theta(a_t|s_t)|_{\theta=\theta_l^{k+1}} \right]$$

where $\theta_0^{k+1} = \theta^k$ and $\theta_{l^*}^{k+1} = \theta^{k+1}$. The step size $\alpha$ is called the learning rate. Special consideration needs to be given to the step size $\alpha$ as its choice affects the learning process, as is the case for any stochastic approximation scheme [93].

To see this, by noting the separation of the reward of a trajectory and the probability of occurrence of a trajectory, we have that the gradient with respect to the policy parameters $\theta$ can be written as

$$\begin{aligned} &\nabla_\theta \sum_{t=1}^{T} \mathbb{E}_{\tau \sim p_{\theta^k}(\cdot)} \left[ e^{\beta R} \log \pi_\theta(a_t|s_t) \right] \\ &\qquad = \sum_{t=1}^{T} \mathbb{E}_{\tau \sim p_{\theta^k}(\cdot)} \left[ e^{\beta R} \nabla_\theta \log \pi_\theta(a_t|s_t) \right] \end{aligned}$$

which follows from the linearity of derivatives.

To economize on the computational cost, in a similar manner as stochastic gradient descent, the true gradient can be approximated by a gradient at a single step, leading to the following

update rule

$$\theta_{l+1}^{k+1} = \theta_l^{k+1} + \alpha \mathbb{E}_{\tau \sim p_{\theta^k}(\cdot)} \Big[ e^{\beta R} \nabla_\theta \log \pi_\theta(a_t|s_t)|_{\theta=\theta_l^{k+1}} \Big].$$

At each iteration of the EM algorithm, to compute the parameter $\theta^{k+1}$, a gradient-based method, starting from some initial value $\theta_0^{k+1} = \theta^k$, iteratively moves in the direction of the gradient, i.e. $\mathbb{E}_{\tau \sim p_{\theta^k}(\cdot)} \Big[ e^{\beta R} \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) \Big]$ until the iterate converges to a value $\theta_*^{k+1}$.

To deal with the nested iteration, one can generate trajectories by following $\theta^k$ and update the $\theta$ for $l = T$ the length of the trajectory, that is

$$\theta_{t+1}^{k+1} = \theta_t^{k+1} + \alpha \mathbb{E}_{\tau \sim p_{\theta^k}(\cdot)} \Big[ e^{\beta R} \nabla_\theta \log \pi_\theta(a_t|s_t)|_{\theta=\theta_t^{k+1}} \Big].$$

This could be further modified and written in terms of reward-to-go

$$\theta_{t+1}^{k+1} = \theta_t^{k+1} + \alpha \mathbb{E}_{\tau \sim p_{\theta^k}(\cdot)} \Big[ e^{\beta R_t} \nabla_\theta \log \pi_\theta(a_t|s_t)|_{\theta=\theta_l^{k+1}} \Big].$$

To see this, note that

$$\begin{aligned}
\theta^{k+1} &= \theta^k + \alpha \mathbb{E}_{\tau \sim p_{\theta^k}(\cdot)} \Big[ e^{\beta R} \nabla_\theta \log \pi_\theta(a_t|s_t) \Big] \\
&= \theta^k + \alpha \mathbb{E}_{\tau \sim p_{\theta^k}(\cdot)} \Big[ e^{\beta R_t^-} e^{\beta R_t} \nabla_\theta \log \pi_\theta(a_t|s_t) \Big] \\
&= \theta^k + \alpha \Big( \mathbb{E}_{\tau \sim p_{\theta^k}(\cdot)} \Big[ e^{\beta R_t^-} \Big] \\
&\qquad \mathbb{E}_{\tau \sim p_{\theta^k}(\cdot)} \Big[ e^{\beta R_t} \nabla_\theta \log \pi_\theta(a_t|s_t) \Big] \Big)
\end{aligned}$$

where $R_t$ is the reward-to-go and $R_t^-$ is the accumulated reward up to time $t$. Note that the first

line is the update rule of (6.10) derived from the EM algorithm. The second line follows from the properties of the exponential function. The last line is due to the temporal structure that leads to the decorrelation between the random variables.

Then by the positivity of the exponential function, we can write

$$\theta^{k+1} = \theta^k + \alpha \sum_{t=1}^{T} \mathbb{E}_{\tau \sim p_{\theta^k}(\cdot)} \left[ e^{\beta R_t} \nabla_\theta \log \pi_\theta(a_t|s_t) \right].$$

Since both the EM and the gradient-based methods are iterative methods, the computation can be reduced by taking a limited number of gradient steps at each iteration of the overall algorithm. Then one can attempt to estimate this expectation with temporal-difference or Monte Carlo methods. The computation of the expectation is expensive. Thus, the update rule of the EM algorithm (6.10) can be re-written,

$$\theta^{k+1} = \theta^k + \alpha e^{\beta R_t} \nabla_\theta \log \pi_\theta(a_t|s_t)$$

Using the full trajectory samples for estimating the reward-to-go $R_t$ leads to a risk-sensitive *Monte-Carlo* algorithm– such an algorithm has been developed in [12] using the MDP framework. A stochastic approximation approach results in the update rule given by

$$\theta^{k+1} = \theta^k + \alpha e^{\beta R_t} \nabla_\theta \log \pi_\theta(a_t|s_t)$$

where an entire trajectory is generated and then $R_t$ samples are used to make an update. This algorithm, as with all Monte Carlo methods, suffers from high variance. Note that, by the equivalence stated in Theorem 10, the log-likelihood optimization of (6.7) when $\beta > 0$ and of (6.8)

when $\beta < 0$ is equivalent to the optimization of the risk-sensitive exponential criteria (3.1). This concludes the proof.

## 6.9 Numerical Example

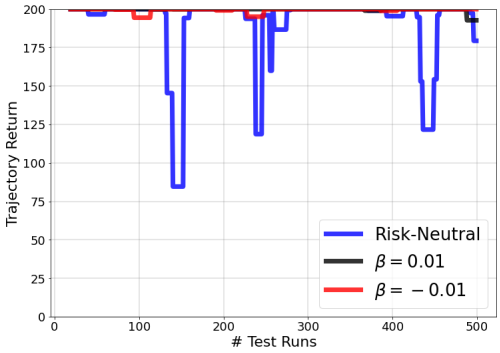



Figure 6.2: The behavior of a trained risk-neutral agent, risk-seeking agent with a risk parameter $\beta = 0.01$ and risk-averse agent with a risk parameter $\beta = -0.01$ during testing in the cart-pole problem. The 0.05-quantile of the trajectory returns, calculated using a moving window of length 20, for 500 independent test runs are plotted.

The fact that the update rules in Theorem 5 is indeed risk-sensitive, in the sense that it considers the tail of the distribution, can be understood from the relation between the log-likelihood and the exponential criteria. To illustrate the risk-sensitivity of the update rule of Theorem 5, we compare the performance of the risk-neutral Monte Carlo policy gradient algorithm, REINFORCE [33], with the risk-sensitive update rule of Theorem 5 on the well-known RL benchmark of Cart-pole (inverted pendulum).

The agent's goal is to balance a pole mounted on a moving cart with an unactuated joint by pushing the cart to the left and right. The state variable consists of the position and velocity of the cart and the angle with the vertical and angular velocity of the pole. The admissible actions are a "left" or "right" force of fixed magnitude. A reward of $+1$ is given for each time step that the pole is kept upright. The agent is successful if it manages to keep the pole upright for 200

time steps and it fails when the pole deviates from the vertical for more than 12 degrees or when the position of the cart is more than 2.4 meters away from the starting point.

The policy is parameterized by a neural network with one hidden layer of 16 neurons and a 'ReLU' activation function. The learning rate is set to $\alpha$=0.01 and the discount factor to $\gamma$=0.99. The optimizer 'Adam' is used with decaying step sizes. We train the agent for 1000 episodes and run an additional 500 test runs.

Both the risk-neutral and risk-sensitive methods are able to learn a policy with an average training reward of close to 200. However, the advantage of risk-sensitive algorithms in this example is highlighted when a measure of the tail of the distribution of the trajectory return over the test runs is considered. For example, out of the 500 test runs, the risk-neutral algorithm failed to keep the pole balanced for more than 150 steps in 10 of the runs, while the risk-averse and risk-seeking algorithms only failed to pass the threshold of 150 steps in 1 and 2 runs, respectively. To better illustrate this point, we use the $\sigma$-quantile of the trajectory return given by $\inf\{x \in \mathbb{R} : P(R \leq x) > \sigma\}$. Fig.6.2 shows the plots of 0.05-quantile of returns. The figure illustrates that the risk-sensitive algorithm (with risk parameter $\beta = \{+0.01, -0.01\}$) achieves higher $\sigma$-quantile (for $\sigma = 0.05$) than the risk-neutral algorithm. The quantile in the figure is calculated using a moving window of length 20 over the returns of the test runs. We observe that the risk-sensitive approach has a noticeable effect on the tail of the distribution.

# Chapter 7: Risk-Sensitive Multi-Agent RL: Independent Learning

## 7.1 Overview

Risk-attitude and trust are interconnected concepts. Risk is always associated with an objective and arises where there is uncertainty. Trust arises where the uncertainty in parts is with regards to other agents' actions. Trust is the extent to which a trusting party is willing to put himself in a vulnerable position with the perception that trustees would not take actions to inflict harm, despite the possibility, that is to say, trust is a level of certainty about the other agent's actions. Thus, trust is a social mechanism that enables cooperation and coordination [112] by reducing the uncertainty and easing the risk associated with other agents' actions.

In the absence of direct communication or formal coordination devices, such as prior agreements, it seems trust is crucial, if not essential, for cooperation and coordination in multi-agent systems. Cooperation and coordination emerge when the agents perceive the risk associated with an action to overcome the cost of collaboration. This perceived risk depends both on who perceives the risk (trusting party) and who is being trusted (trustee), that is to say, intuitively, the emergence of collaboration and coordination depends on the risk-attitudes of both the trusting party and the trustee(s).

The role of risk-attitudes in the emergence of coordination has been investigated in ad-hoc case-by-case human-subject studies in the literature. For example, [113] investigates the impact

of information about the risk-attitude of an opponent on an agent's action in Stag-Hunt games using human-subject experiments. Computational models of decision making that model the agent's risk-attitude allow for a systematic study of such relations. Risk-sensitive Reinforcement Learning is such a computational model of decision making.

Classical (risk-neutral) Reinforcement Learning (RL) [6] aims to optimize a long-run expectation of a desired performance measure, e.g. cumulative (un) discounted return. Risk-sensitive RL algorithms incorporate some notion of risk, e.g. some functions of higher moments of the desired performance measure, such as Conditional Value at Risk [24], Variance [23, 26] and exponential (exponential of Integral) criteria [27, 114].

Risk-sensitive REINFORCE [114] is a Risk-sensitive Policy Gradient RL algorithm that models the agent's risk-attitude by leveraging the exponential criteria [43]. The agent's risk-attitude in Risk-sensitive REINFORCE is controlled by a single real-valued parameter $\gamma$ where an agent with $\gamma<0$ exhibits risk-averse behavior and with $\gamma>0$ risk-seeking behavior. The larger magnitude of the risk-parameter $\gamma$ results in a more risk-sensitive (averse/seeking) behavior and an agent shows risk-neutral behavior as $\gamma$ goes to zero. With this parameter as a knob for adjusting the risk-attitude of the agent, we aim to study the effects of risk-attitude on the emergence of coordination in multi-agent decision-making.

To focus on the role of risk-attitudes on the emergence of coordination and to isolate the effects of communication and coordination devices, we investigate multi-agent environments with independent agents, that is to say, the agents do not communicate directly and have no priors over other agents' behaviors; treating other agents as part of the environment.

An analysis of a learning algorithm in multi-agent environments needs to answer (I) the question of Stability — whether or not the algorithm converges (usually to a Nash equilibrium) —

and (II) the question of Equilibrium Selection [115, 116], that is, where the algorithm converges to. Nash equilibrium as a solution concept has been the focus of numerous Multi-agent RL algorithms. Although Nash equilibrium is a well-justified solution concept for the analysis of Human-centered systems, it might not be Hicks optimal (Hicks optimal outcome is one that the total payoff for all of the agents of a game is the most it could be) nor Pareto optimal (Pareto optimal outcome is one that no agent can be better off without making at least one other agent worse off) and therefore might not be the best solution concept for the design of an artificial multi-agent system.

We explore the connection between agents' risk-attitudes and the emergence of coordination using Risk-sensitive RL as the computational model of decision making. To that end, we study the role of risk-attitudes in repeated 2-agent coordination games, e.g. Stag-Hunt (trust-dilemma) [117], using Independent Risk-sensitive REINFORCE RL-agents. Such simple games capture the intuition behind risk-sensitive methods in multi-agent environments. The study of social games to investigate the emergence of stable mutual cooperation has a long history [118]. We show that Risk-sensitive Policy Gradient agents with the "right" attitudes (appropriately chosen risk-parameter $\gamma$) can achieve the Hicks (Pareto) optimal equilibrium, i.e. the highest payoff possible, where (risk-neutral) Policy Gradient fails to find the Hicks optimal equilibrium. That is, we experimentally show that risk-attitude shapes the basin of attraction of the equilibriums. Our simulation results confirm that agents' risk-attitudes not only can facilitate coordination and influence the learned behavior (where the algorithm converges to), but also influence the convergence speed of the algorithm, and in turn the algorithm's sample efficiency.

## 7.2 Related Work

Such Independent Learning algorithms have been investigated in the literature [119, 120, 121, 122]; the existing Multi-agent RL algorithms do not model agents' risk-attitudes. Nash equilibrium as a solution concept has been the focus of numerous Multi-agent RL algorithms [123, 124, 125, 126, 127, 128, 129]

## 7.3 Coordination Games

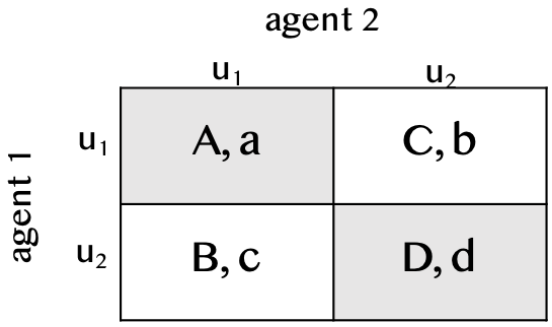



Figure 7.1: The payoff matrix of a generic coordination game

Coordination game is a class of simultaneous finite games where coordination with the other agent(s) yields a better payoff for all agents. A 2-agent (and 2-action) Coordination game has multiple pure Nash equilibriums given by a matrix form as shown below where for agent 1 (rows) A>B, D>C, and for agent 2 (columns): a>b, d>c.

In this game, the strategy profiles (A,a) and payoff (D,d) are pure Nash equilibriums. In such a 2-agent matrix game, the collective behavior of the agents are characterized by their joint policy $\pi$:=$(\pi^1, \pi^2)$ where the agents' individual policies $\pi^1$=$(\alpha, 1-\alpha)$ and $\pi^2$=$(\beta, 1-\beta)$ characterize each agent's behavior in the multi-agent environment. Note that $\alpha$ and $\beta$ are the probability with which agents 1 and 2 choose action $u_1$, i.e. $Pr(a^1 = u_1)$=$\alpha$ and $Pr(a^2 = u_1)$=$\beta$.

Also note that for pure strategies, the policy is deterministic, i.e., $\alpha$=0 or $\alpha$=1 (resp. $\beta$=0 or $\beta$=1). The mixed Nash equilibrium $\pi_{mix}$:=$(\pi^1_{mix}, \pi^2_{mix})$ is given by the stochastic policies $\pi^1_{mix}$=$(\alpha^{mix}, 1-\alpha^{mix})$ and $\pi^2_{mix}$=$(\beta^{mix}, 1-\beta^{mix})$, where

$$\alpha^{mix} = \frac{d-c}{a+d-b-c}, \quad \beta^{mix} = \frac{D-C}{A+D-B-C} \tag{7.1}$$

Under the assumption of full support, the expected utility of agent 1, $J^1(\pi)$, and agent 2, $J^2(\pi)$, by following the joint policy $\pi$, is given by

$$J^1(\pi) = \alpha\beta A + \alpha(1-\beta)C + (1-\alpha)\beta B + (1-\alpha)(1-\beta)D \tag{7.2}$$

$$J^2(\pi) = \alpha\beta a + \alpha(1-\beta)b + (1-\alpha)\beta c + (1-\alpha)(1-\beta)d \tag{7.3}$$

### 7.3.1 Stag-Hunt

Classical example of a coordination game is Stag-Hunt (Trust-dilemma) game [117]: two hunters need to decide independently, that is, in the absence of direct communication and coordination devices, to either track a stag (action $u_1$ (Stag)) or hunt a hare (action $u_2$ (Hare)). Hunting a stag has the highest payoff, but requires both hunters to track the stag together. If one decides to track the stag and his counterpart goes for hunting a hare he would not be able to hunt the stag and would end up with no payoff, hence choosing Stag is a risky action with a potentially high payoff. On the other hand, hunters could hunt a hare without the help of their counterpart and would receive a payoff less than what they would have received for a stag, hence choosing Hare is a safe action with a relatively low payoff.

This game is a 2-agent coordination game and has a coordination game payoff matrix with

$A = a$, $B = b$, $C{=}c$, and $D = d$, where $a{>}b{\geq}d{>}c$. The Stag-Hunt game has two pure Nash equilibriums – one that is risk-dominant $(d, d)$ (Hare-Hare) and another that is payoff-dominant $(a, a)$ (Stag-Stag) – and a mixed Nash equilibrium (c.f. Eq. (7.1)) given by

$$\alpha^* = \alpha^{\text{mix}} = \beta^{\text{mix}} = \frac{d - c}{a + d - b - c} \tag{7.4}$$

The Stag-Hunt is a game that describes a conflict between safety and cooperation. The strategy pair (a,a) is payoff-dominant since payoffs are higher for both agents compared to the other pure Nash equilibrium, (d,d). The strategy pair (a,a) is Hicks optimal, and therefore Pareto optimal. On the other hand, (d,d) is risk-dominant, since, despite the other agent's action, it provides a higher expected payoff. The more uncertainty agents have about the actions of the other agent(s) (less trust), the more likely they will choose the strategy corresponding to the risk-dominant equilibrium. The mixed Nash equilibrium falls in between the two pure Nash equilibriums in terms of payoff and safety. The lower values of c induce more risk in the Pareto optimal equilibrium.

### 7.3.2 Repeated Games and Learning

We consider matrix games in an episodic setting in which both agents repeatedly, for a fixed number of rounds $T$ per episode, execute arbitrary policies for a fixed number of episodes. Repeated games allow the agents to adjust their policies at each playing round so as to find (learn) a policy that maximizes their desired performance measure. We explore the behavior of Risk-neutral and Risk-sensitive Policy Gradient RL algorithms in such repeated games.

## 7.4 Independent Risk-sensitive Policy Gradient in Multi-agent Reinforcement Learning

An analysis of learning algorithms in multi-agent environments ought to answer two basic questions: first is that of Stability which considers whether or not the algorithm converges (usually to a Nash equilibrium) and second is the question of Equilibrium Selection which concerns where (which equilibrium) the algorithm converges to. We experimentally show, in a repeated Stag-Hunt game, Risk-sensitive Policy Gradient RL-agents, with appropriately chosen risk parameters, can converge (very quickly) to the Pareto optimal equilibrium where Policy Gradient algorithms fail to find such equilibrium. We experimentally observed that all risk parameters greater than a threshold value, $\gamma > \gamma^*$ led to fast convergence to the Hicks optimal equilibrium. Thus, finding an appropriate risk-parameter proved to be an easy task.

We study independent Risk-sensitive RL-agents in a finite repeated game (2-agent Stag-Hunt), where each agent independently (in the absence of direct communication and coordination devices) selects an action and observes only their own payoff and the joint action that was taken previously; the agents do not know the risk-attitudes of the other agents nor their policies. In the remainder of this section, we first introduce some notations and then present Independent Learning with standard (risk-neutral) REINFORCE [86] and risk-sensitive REINFORCE [114] Policy Gradient RL-agents.

### 7.4.1 Stochastic Games

Repeated Matrix games, e.g. repeated 2-agent Stag-Hunt, are full-state Stochastic (Markov) games [130]. In a repeated 2-agent Stag-Hunt game, at each playing round in an episode, agent $i \in \{1, 2\}$ observes the state $s_t \in S$, where $S$ is the state-space, and executes an action $a_t^i \in U = \{u_1(Stag), u_2(Hare)\}$, where $U$ is the agent's action space, according to its policy $\pi^i$, a distribution over actions given state (possibly a smooth parametrized function, such as a Neural Network with parameters $\theta^i \in \mathbb{R}^d$). Upon the execution of the joint action $a_t = (a_t^1, a_t^2)$, the system transitions to a successor state $s_{t+1}$. Matrix games, e.g. (repeated) Stag-Hunt, allow the agents to make their policy conditioned on the past actions, e.g. Tit-for-tat strategies [118], which leads to concepts such as reputation, retaliation, and trust. We consider a stochastic game with the joint action taken at the previous playing round as the state, i.e., $s_t = (a_{t-1}^1, a_{t-1}^2)$. Thus, in a repeated Stag-Hunt game, for Independent agents, that is, agents that treat the other agent(s) as part of the environment, the system transition probabilities, which prescribe the probability with which the state transitions to $s_{t+1}$ given the current state $s_t$ and the joint action $a_t$, are only a function of the other agents' policies. At each playing round, each agent i receives an instantaneous payoff $r_t^i(a_t^i, a_t^{-i}) \in [r_{min}, r_{max}]$.

The agents start with some initial parameters $\theta^i$ (corresponding to some initial Stag probability $\alpha$ and $\beta$), having observed the joint action (the choice of other agents) in the previous round, the agents modify their policy (their Stag probability) according to their algorithms as to optimize some desired performance measure.

### 7.4.2 Risk-neutral Policy Gradient

In classical (risk-neutral) Independent RL, the desired performance measure to be optimized is usually an expectation of some log-run objective. A common example of such objective in RL literature is expected (undiscounted) cumulative Return, that is, each agent $i$ solves

$$\max_{\theta^i} J^i(\theta) := \mathbb{E}_\pi\Big[R^i\Big], \tag{7.5}$$

where $R^i = \sum_{t=0}^{T-1} \zeta^t r_t^i$ is the agent $i$ cumulative payoff over an episode (T playing rounds), $\zeta \in (0, 1]$ is a discount factor, and $\theta := (\theta^1, \theta^2)$, that is, an agent's performance is coupled with the policy of the other agent(s) in the environment; the expectation is taken with respect to the stochastic policies.

Classical Policy Gradient algorithms are such risk-neutral algorithms which use an iterative gradient ascent to find the optimal policy

$$\theta_{t+1}^i = \theta_t^i + \eta \widehat{\nabla J^i(\theta_t)} \tag{7.6}$$

where $\eta \in \mathbb{R}$ is a constant step-size, i.e., learning rate, and $\widehat{\nabla J^i(\theta_t)} \in \mathbb{R}^d$ denotes a stochastic estimate of the gradient of the system's performance measure with respect to the agents' policy parameter $\theta^i$. Standard REINFORCE [86] algorithm is such Gradient algorithm.

In a repeated Stag-Hunt game, (risk-neutral) Policy Gradient (PG) algorithms [6] converge to one of the pure Nash equilibriums depending on the agents' initial policies, and the payoff matrix. The agents start with some initial policy (Stag probability $\alpha$ and $\beta$), having observed

the joint action (the choice of other agents) in the previous round, the agents modify their behavior by changing their policy parameters according to their learning algorithm. If both agents' initial policies take the risky action (Stag) with a frequency higher than the mixed equilibrium, i.e. $\alpha, \beta > \alpha^*$, then the Policy Gradient agents will converge to the payoff-dominant equilibrium (Stag,Stag), and, similarly, if both agents take the safe action (Hare) with a frequency lower than the mixed equilibrium, i.e. $\alpha, \beta < \alpha^*$, then the agents converge to the risk-dominant equilibrium (Hare,Hare). That is to say, $\alpha, \beta > \alpha^*$ fall within the basin of attraction of the payoff-dominant equilibrium (Stag,Stag) and $\alpha, \beta < \alpha^*$ falls within the basin of attraction of the risk-dominant equilibrium (Hare,Hare) [131].

**Theorem 6.** *[132] Let $\alpha(0)$ and $\beta(0)$ be the initial Stag probability of agents 1 and 2, respectively, and $\alpha^*$ be the mixed Nash equilibrium Stag probability of the game. Then the (risk-neutral) Policy Gradient is able to discover the high-payoff Nash Equilibrium if the initial policies are set such that $\alpha(0) > \alpha^*$ or $\beta(0) > \alpha^*$.*

*Proof.* To see this, note that

$$\nabla_\alpha J^1(\pi) = \beta a + (1-\beta)c - \beta b - (1-\beta)d$$

$$\nabla_\beta J^2(\pi) = \alpha a - \alpha c + (1-\alpha)b - (1-\alpha)d$$

In order for the algorithm to find the (A,a) equilibrium, both $\alpha$ and $\beta$ need to increase, i.e. $\nabla_\alpha J(\pi), \nabla_\beta J(\pi) > 0$, thus,

$$\alpha, \beta > \frac{d-c}{a+d-c-b}$$

Recall that the mixed Nash Equilibrium $\alpha^*=(d-c)/(a+d-b-c)$. □

**Corollary 11.1.** *[131] Suppose* $a-b=\epsilon(d-c)$ *for some* $0<\epsilon<1$ *and initialize the agents polices uniformly, i.e.,* $\alpha(0),\beta(0)\sim U[0,1]$. *Then, the probability that (risk-neutral) Policy Gradient discovers the high-payoff Nash Equilibrium is upper bounded by* $\frac{2\epsilon+\epsilon^2}{1+2\epsilon+\epsilon^2}$.

### 7.4.3 Risk-sensitive Policy Gradient

Risk-sensitive REINFORCE algorithm [114], a risk-sensitive variant of standard (risk-neutral) REINFORCE algorithm, takes into account the risk-attitude of an agent by leveraging the exponential criteria [43]. Thus, Independent Risk-sensitive REINFORCE RL-agents each seeks a stochastic policy $\pi^i$, so as to maximize the exponential criteria, that is, each agent i solves

$$\max_{\theta^i} J^i_{\gamma^i}(\theta) := \mathbb{E}_\pi\left[e^{\gamma^i R^i}\right] \tag{7.7}$$

where $R^i=\sum_{t=0}^{T-1}\zeta^t r^i_t$ is the agent's cumulative payoff (reward) over an episode (T playing rounds) and $\zeta\in(0,1]$ is a discount factor. Thus, the optimal policy parameter can be obtained iteratively by using gradient ascent as follows

$$\theta^i_{t+1}=\theta^i_t+\eta\widehat{\nabla J^i_{\gamma^i}(\theta_t)} \tag{7.8}$$

where $\eta\in\mathbb{R}$ is a constant step-size, i.e., learning rate, and $\widehat{\nabla J^i_{\gamma^i}(\theta_t)}\in\mathbb{R}^d$ denotes a stochastic estimate of the gradient of the system's performance measure with respect to the agent i policy parameter.

The gradient of the risk-sensitive objective of (2.2) with respect to the policy parameters of

agent i is given by Risk-sensitive Policy Gradient theorem [114] as

$$\nabla J^i_{\gamma^i}(\theta) \propto \mathbb{E}_{\pi,P}\Big[\sum_{t=0}^{T-1} \gamma^i e^{\gamma^i R^i} \nabla \log \pi^i(a^i_t|s_t;\theta^i)\Big] \tag{7.9}$$

where $R^i := \sum_{t'=1}^{T-1} \zeta^{t'} r^i_{t'}$. Then the update rule that an Independent Risk-sensitive REINFORCE RL-agent applies at each playing round is given by

$$\theta^i_{t+1} = \theta^i_t + \eta\gamma^i e^{\gamma^i R^i_t} \nabla \log \pi^i(a^i_t|s_t;\theta^i) \tag{7.10}$$

**Remark 17.** *To see how the exponential criteria take risk into account, note that the exponential criteria may be written in Taylor's series as,*

$$\mathbb{E}\Big[\gamma e^{\gamma R}\Big] = \gamma + \gamma^2 \mathbb{E}\Big[R\Big] + \frac{\gamma^3}{2}\mathbb{E}\Big[R^2\Big] + \dots$$

*The parameter $\gamma$ controls the risk-sensitiveness of the objective, i.e., trade-offs the maximization of the expectation and minimization of risk. An agent with risk-parameter $\gamma<0$ exhibits risk-averse behavior and with $\gamma>0$ risk-seeking behavior. One can see that by noting the equivalent game formulation of exponential criteria. The agent becomes risk-neutral as $\gamma$ approaches to 0, i.e., $\gamma\to 0$, that is, $\lim_{\gamma\to 0} J_\gamma(\theta) = J(\theta)$.*

## 7.5 Numerical Experiments and Simulation

We aim to investigate the role of risk-attitudes in the emergence of coordination by answering the following questions:

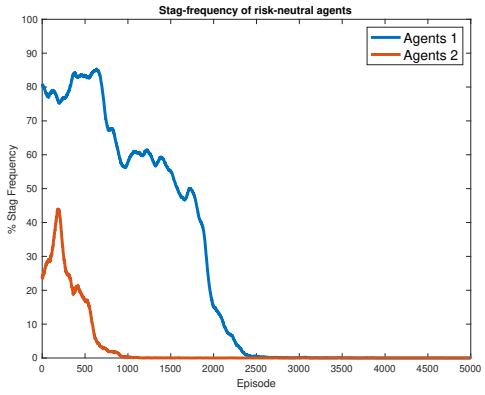



Figure 7.2: The (risk-neutral) REINFORCE agents converge to the risk-dominant (low payoff) Nash Equilibrium and fail to find the Hicks optimal equilibrium.

- Does the introduction of risk-sensitive agents promote cooperation and coordination?
- Does the introduction of risk-sensitive agents improve the agents' sample efficiency?
- What qualitative conclusions can be drawn about the risk-attitudes, equilibrium selection, and optimality?

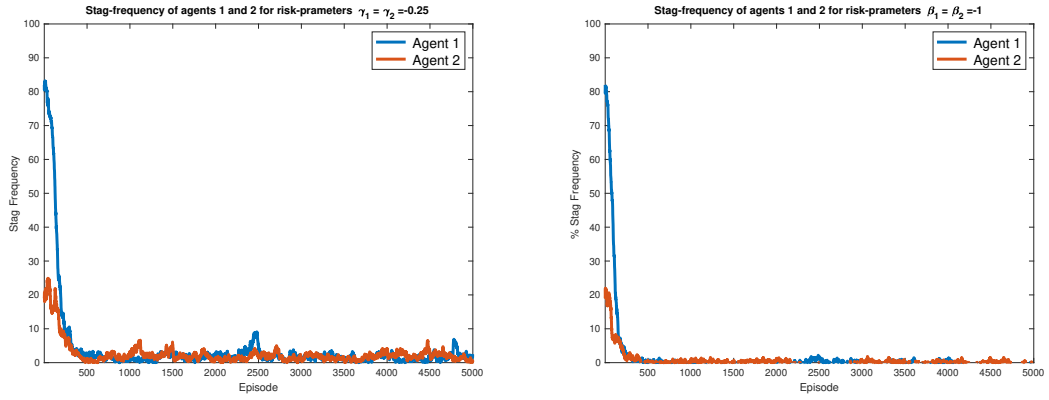
To that end, we consider (Risk-sensitive/Risk-neutral) REINFORCE RL-agents in an instance of a repeated 2-agent Stag-Hunt game. Given a set of initial policies and a payoff matrix, we experimentally investigate the role of Risk-sensitive REINFORCE agents on the learning dynamics and emergence of coordination in a repeated 2-agent Stag-Hunt game.

### 7.5.1 Stage-Hunt Numerical Instance

We consider an instant of Stag-Hunt game with a payoff matrix given by  In this game, the pure Nash equilibriums are the payoff-dominant $(5,5)$ and the risk-dominant $(2,2)$ strategy profiles. The mixed strategy Nash equilibrium $\pi_{\text{mix}}=(\pi^1_{\text{mix}},\pi^2_{\text{mix}})$ where $\pi^1=\pi^2=(2/3,1/3)$, that is to say, each hunter chooses Stag with probability $2/3$ and Hare with probability $1/3$. The

| agent 1 \ agent 2 | $u_1$ | $u_2$ |
|---|---|---|
| $u_1$ | 5, 5 | 0, 4 |
| $u_2$ | 4, 0 | 2, 2 |

Figure 7.3: A numerical example of 2-agent Stag-Hunt Game

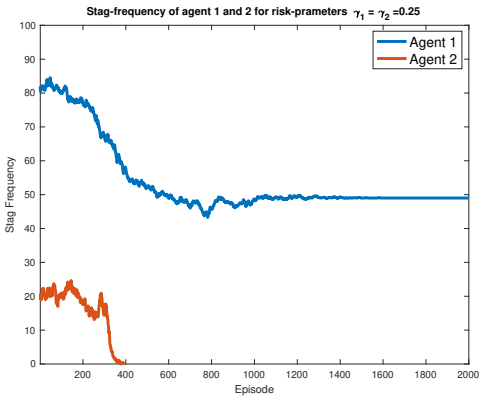



(a) $\gamma_1 = \gamma_2 = -0.25$ (b) $\gamma_1 = \gamma_2 = -1$

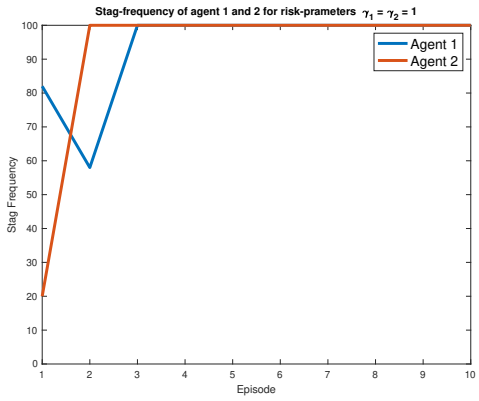
Figure 7.4: Risk-averse agents in repeated Stag-Hunt game: Stag-frequency played by agent 1 and agent 2

payoffs of the mixed Nash equilibrium is $(J^1(\pi) = J^2(\pi)) = 10/3$.

The agents' policies are parameterized by simple Neural Networks with a 2-neuron input layer, a 2-neuron fully connected hidden layer, and a Softmax output layer; all hyper-parameters, such as learning rate, gradient threshold, etc, are left as the default values in the simulation platform (here `MATLAB` - Simulink). We trained (Risk-sensitive/Risk-neutral) Policy Gradient agents for several different values of the risk-parameter $\gamma$. The agents play 100 rounds in each episode. Figures show the Stag frequency for each agent during the learning episodes (smoothed over a window of length 10) which characterizes the agents' learning dynamics.




(a) $\gamma_1$=$\gamma_2$=0.25




(b) $\gamma_1$=$\gamma_2$=1

Figure 7.5: Risk-seeking agents in repeated Stag-Hunt game: Stag-frequency played by agent 1 and agent 2

### 7.5.2 Results and Discussion

Policy Gradient fails to find the Hicks optimal equilibrium and converges to the risk-dominant (safe) equilibrium (Hare, Hare). Note that the agents' initial policies are such that agent 1 has an initial Stag probability greater than the Stag probability of the mixed Nash and agent 2 has a Stag frequency less than the Stag probability of the mixed Nash equilibrium. Thus, by Theorem 6, we expect the (risk-neutral) Policy Gradient agents to converge to the risk-dominant Nash equilibrium of Hare-Hare with high probability. Figure 7.2 shows the policy (Stag frequency) evolution throughout the training for two risk-neutral REINFORCE Policy Gradient agents (without a baseline). The risk-neutral agents converge to the risk-dominant (low-payoff) Nash equilibrium by the 2500th episode.

The risk-averse agents converge to the risk-dominant (safe) equilibrium (Hare, Hare) with a convergence speed propositional to the magnitude of the risk-aversion parameter. Figure 7.4 shows the Stag-frequencies for two risk-averse agents (with the same risk-aversion parameters). The larger magnitudes of the risk-aversion parameter (up to a saturation value) result in faster

convergence to the risk-dominant (safe) Nash equilibrium, and in turn, improve the agents' sample efficiency. One can see this, by noting how fast the plots in Figure 7.4 (b) ($\gamma_1$=$\gamma_2$=$-1$) drop to the proximity of zero as compared to Figure 7.4 (a) ($\gamma_1$=$\gamma_2$=$-0.25$).

The risk-seeking agents with a risk-parameter above a threshold converge to the Hicks optimal equilibrium (payoff-dominant) Nash equilibrium. Figure 7.5 shows the Stag-frequencies for two risk-seeking agents (with the same risk-seeking parameters). The risk-seeking agents with a risk-parameter above a threshold converge to the payoff-dominant (unsafe) Nash equilibrium, but for the risk-parameter between 0 to the threshold, e.g. $\gamma$=$0.25$, the agents show a different behavior; the agent with the higher initial Stag-frequency converges to Stag frequency 1/2 and the agent with the lower initial Stag-frequency converges to 0 stag frequencies. The conclusion is that "sufficiently" risk-seeking agents converge to the Hicks Nash equilibrium, however, risk-seeking but not ambitious enough agents (risk-seeking with low magnitude) converge to a non-Nash solution.

We performed a series of simulations with a mix of risk-averse and risk-seeking agents and with different magnitudes for the risk-parameters which we have not included due to space limitations. The analysis of the simulation results with such mixed agents reaffirms the results obtained for the case of identical agents, that is, where the agents converge and how fast depends on the agents' risk-attitudes (the sign of $\gamma^i$'s and their magnitude. The appropriate risk-parameter for the convergence to the Hicks optimal equilibrium depends on the risk-attitude of all agents, as well as their initial policies, and the payoff matrix.

Part III

# Beyond RL: Risk in Decision Making

In this study, we aim to investigate the concept of risk-sensitivity and the robust properties of risk-sensitive algorithms in decision-making and optimization domains beyond the realm of Reinforcement Learning (RL). Specifically, we focus on Safe Reinforcement Learning utilizing risk-sensitive filters. Through our exploration, we seek to enhance the understanding and applicability of risk-sensitive approaches in various domains.

# Chapter 8: Risk-Sensitive Safety Filters

The material presented in this chapter is based on a joint work conducted by Armin Lederer and Sandra Hirche [133].

## 8.1 Overview

Having a pause before responding is a mental technique that helps humans perceive, control, and manage our emotions. Human's ability to think before reacting, especially in difficult and complex situations, is a cognitive mechanism to keep our actions in check. This cognitive process is called inhibitory control, also known as response inhibition [134]. Response inhibition allows an individual to inhibit their prepotent (natural and habitual) responses in order to select a more appropriate (e.g. safer) behavior.

## 8.2 Related Work

Independent from this foundation in psychology, response inhibition has become increasingly popular in learning-based control [135] and Reinforcement Learning (RL) [136] in recent years, where safety is a major concern [137]. The idea is to decouple optimality and safety by independently determining safe and optimal control laws. Before applying an optimal, but potentially unsafe control input to the real system, its safety is checked, such that a safe control input

can be chosen instead [138]. Thereby, the prepotent optimal response is inhibited to guarantee the safety of the closed-loop system.

The challenge of this approach lies in finding safe policies and efficient methods to determine the safety of a control input online. When the dynamics of the systems are known to exhibit a control-affine structure, control barrier functions (CBF) can be effectively employed to address this challenge [139]. Since their analytical derivation for more flexible classes of dynamical systems is difficult at best, techniques from model predictive control have become popular for computing safe backup strategies online [140, 141]. While such predictive safety filters provide a conceptionally flexible approach for realizing inhibitory control, they generally suffer from high computational complexity. This limitation can be mitigated by combining ideas from reachability analysis [142] or optimal control [143] with reinforcement learning techniques to learn safety conditions and safe control laws offline, such that resource-demanding computations can be avoided during the application of the inhibited control law.

While these approaches allow the seemingly straightforward realization of inhibitory control for ensuring the safety of real-world systems, they do not consider the risk of losing safety due to uncertainty arising from approximate system models and process noise. This is in strong contrast to humans, for which psychological studies have shown a critical link between response inhibition and an individual's risk attitude (willingness to take risk or not) [144]. When inhibitory control is implemented in technical systems through analytically derived safety conditions such as CBFs, this risk-sensitivity can be easily achieved by reformulating standard conditions using risk measures [145, 146]. However, the extension to flexible approaches for constructing safety conditions, e.g., using RL techniques remains an open problem.

## 8.3 Problem Statement

We consider a discrete-time dynamical system[1]

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\omega}_k), \tag{8.1}$$

where $\mathbf{x}_k \in \mathbb{X} \subset \mathbb{R}^{d_x}$ are states, $\mathbf{u}_k \in \mathbb{U} \subset \mathbb{R}^{d_u}$ are control inputs, $\boldsymbol{\omega}_k \in \Omega \subset \mathbb{R}^{d_\omega}$, $\boldsymbol{\omega}_k \sim \rho$ is i.i.d. process noise drawn from some distribution $\rho$ with zero mean, and $\mathbf{f} : \mathbb{X} \times \mathbb{U} \times \Omega \to \mathbb{X}$ denotes an unknown, continuous transition function. We assume that a nominal, potentially unsafe policy $\boldsymbol{\pi}^* : \mathbb{X} \to \mathbb{U}$ is given, which can be obtained, e.g., using standard reinforcement learning techniques [136].

The goal is to render the nominal policy safe using inhibitory control of the form

$$\boldsymbol{\pi}^*_{\text{safe}}(\mathbf{x}) = \arg\min_{\mathbf{u} \in \mathbb{U}} \|\boldsymbol{\pi}^*(\mathbf{x}) - \mathbf{u}\| \tag{8.2a}$$

$$\text{such that } \mathbf{u} \text{ is safe.} \tag{8.2b}$$

In this response inhibition, our notion of safety follows the common principle of classifying the state space $\mathbb{X}$ into a safe region $\mathbb{X}_{\text{safe}} \subset \mathbb{X}$ and an unsafe region $\mathbb{X}_{\text{unsafe}} = \mathbb{X} \setminus \mathbb{X}_{\text{safe}}$. For example, the safe set $\mathbb{X}_{\text{safe}}$ can represent the joint angles for which self-collisions of a robotic manipulator are excluded. Due to the process noise $\boldsymbol{\omega}$ with a potentially unbounded probability distribution, it is generally not possible to deterministically ensure that the system never enters the unsafe state space $\mathbb{X}_{\text{unsafe}}$. Therefore, we define safety probabilistically through the following

[1] Notation: Lower/upper case bold symbols denote vectors/matrices, blackboard bold letters denote sets, $\mathbb{R}_+/\mathbb{R}_{0,+}$ all real positive/non-negative numbers, $\|\cdot\|$ the Euclidean norm, $\mathbb{E}_x[\cdot]$ the expectation with respect to the distribution of $x$, and $\mathbb{P}(\cdot)$ the probability.

form of forward invariance.

**Definition 2.** *A policy $\pi(\cdot)$ is called $\delta$-safe if there exists a subset $\mathbb{V} \subseteq \mathbb{X}_{\text{safe}}$ such that*

$$\mathcal{P}(\mathbf{f}(\mathbf{x}, \pi(\mathbf{x}), \omega) \in \mathbb{V}) \geq 1 - \delta \forall \mathbf{x} \in \mathbb{V}.$$

Since definition 2 requires a form of forward invariance of $\mathbb{V}$, it immediately induces guarantees for all states along a K-step trajectories of the form

$$\mathcal{P}(\mathbf{x}_k \in \mathbb{V},\ \forall k = 1 \ldots, K) \geq (1 - \delta)^K, \tag{8.3}$$

where $\mathbf{x}_k$ is defined through iterative application of (8.1). Hence, the considered notion of safety in this paper is stronger than merely requiring the next state to lie in the safe subspace, i.e., $\mathcal{P}(\mathbf{f}(\mathbf{x}, \pi(\mathbf{x}), \omega) \in \mathbb{X}_{\text{safe}}) \geq 1 - \delta$.

Based on the definition of $\delta$-safety, we consider the problem of deriving a tractable safety condition (8.2b) for inhibitory control, which is guaranteed to be feasible for some risk-aversion as measured through $\delta$. Since we assume the transition function $\mathbf{f}$ is unknown, solving this problem is generally impossible without any further assumptions. Therefore, we require the availability of a probabilistic model in the form of a distribution over functions as formalized in the following.

**Assumption 1.** *A probability distribution $\mathcal{F}$ over potential dynamics $\mathbf{f}$ is known, i.e., $\mathbf{f} \sim \mathcal{F}$.*

In practice, suitable distributions over functions $\mathcal{F}$ can be straightforwardly obtained using Bayes' theorem, e.g., through Gaussian process regression [147]. Moreover, approximate distri-

butions can be learned using deep ensembles [148]. Therefore, this assumption is not restrictive in practice.

## 8.4 Risk-Sensitive Inhibitory Control

Even with the knowledge of $\mathcal{F}$, determining a safety condition (8.2b) is a challenging problem since we generally do not know which subset $\mathbb{V}$ is suitable for definition 2. Here, we follow the ideas of [143] and employ RL techniques to define these subsets through a value function. For this purpose, we first show how state constraints can be expressed through risk-sensitive cost conditions in section 8.4.1. After deriving these safety conditions, in section 8.4.2, we address the problem of learning a separate, so-called backup policy whose pure focus lies on ensuring safety. Based on this policy, a risk-sensitive safety filter for realizing inhibitory control in reinforcement learning is finally presented in section 8.4.3.

### 8.4.1 State Constraints as Risk-Sensitive Cost Conditions

In order to express state constraints through risk-sensitive cost conditions, we define the expected cumulative cost for a policy $\boldsymbol{\pi}(\cdot)$ as

$$V_{\boldsymbol{\pi}}(\mathbf{x}) = \mathbb{E}_{f,\omega}\left[\sum_{k=0}^{\infty} \gamma^k c(\mathbf{x}_k)\right], \tag{8.4}$$

where $c : \mathbb{R}^{d_x} \to \mathbb{R}_{0,+}$ denotes an immediate cost and $\mathbf{x}_k$ is defined through the iterative application of (8.1) with $\mathbf{x}_0 = \mathbf{x}$ and $\mathbf{u}_k = \boldsymbol{\pi}(\mathbf{x}_k)$. If the immediate cost $c(\cdot)$ can be used as an indicator of the unsafe subset $\mathbb{X}_{\text{unsafe}}$, there exists a sub-level set of $V_{\boldsymbol{\pi}}(\cdot)$ contained in $\mathbb{X}_{\text{safe}}$, as guaranteed by the following lemma.

**Lemma 12** ([143]). *Assume there exists a constant $\hat{c} \in \mathbb{R}_+$, such that the cost $c : \mathbb{R}^{d_x} \to \mathbb{R}_{0,+}$ satisfies*

$$c(\mathbf{x}) \geq \hat{c} \quad \forall \mathbf{x} \in \mathbb{X}_{\text{unsafe}}. \tag{8.5}$$

*Then, there exists a constant $\bar{\xi} \in \mathbb{R}_+$, such that the intersection between the sub-level set $\mathbb{V}_\pi^{\bar{\xi}} = \{\mathbf{x} \in \mathbb{X} : V_\pi(\mathbf{x}) \leq \bar{\xi}\}$ and $\mathbb{X}_{\text{unssafe}}$ is empty, i.e., $\mathbb{V}_\pi^{\bar{\xi}} \cap \mathbb{X}_{\text{unsafe}} = \emptyset$.*

Based on this lemma, we can choose any sub-level set $\mathbb{V}_\pi^{\xi}$ with $\xi \leq \bar{\xi}$ for showing $\delta$-safety as introduced in definition 2. Therefore, it only remains to derive conditions that ensure the state stays in $\mathbb{V}_\pi^{\xi}$ after a transition. While this could be achieved using a probabilistic "worst case" consideration as shown in [143], this approach yields a computationally challenging min-max problem for unknown system dynamics. Therefore, we follow a fully probabilistic approach by introducing the risk operator [149]

$$\mathbb{R}_\beta[C] = \frac{1}{\beta} \log\left(\mathbb{E}\left[\exp\left(\beta C\right)\right]\right) \tag{8.6}$$

for an arbitrary random variable $C$ and risk parameter $\beta \in \mathbb{R}_+$. This operator allows the derivation of a computationally efficient condition for ensuring $\delta$-safety as shown in the following proposition.

**Proposition 3.** *Consider a cost function $c(\cdot)$ satisfying (8.5). If there exist constants $\xi, \beta \in \mathbb{R}_+$ with $\xi < \bar{\xi}$ such that*

$$\mathbb{R}_\beta[V_\pi(\mathbf{x}^+)] \leq \xi, \qquad \forall \mathbf{x} \in \mathbb{V}_\pi^{\bar{\xi}} \tag{8.7}$$

*holds for* $\mathbf{x}^+ = \mathbf{f}(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x}), \boldsymbol{\omega})$*, then,* $\boldsymbol{\pi}(\cdot)$ *is* $\delta$*-safe on* $\mathbb{V}_{\boldsymbol{\pi}}^{\xi}$ *with*

$$\delta = \exp\left(\beta\left(\xi - \bar{\xi}\right)\right). \tag{8.8}$$

*Proof.* Due to lemma 12, we can bound the probability of leaving $\mathbb{X}_{\text{safe}}$ by the probability of leaving $\mathbb{V}_{\boldsymbol{\pi}}^{\bar{\xi}}$. Therefore, it is sufficient to derive an upper bound for the probability

$$\mathbb{P}\left(V_{\boldsymbol{\pi}}(\mathbf{x}^+) \geq \bar{\xi}\right) = \mathbb{E}_{\mathbf{x}^+}\left[I_{\bar{\xi}}(V_{\boldsymbol{\pi}}(\mathbf{x}^+)\right], \tag{8.9}$$

where the indicator function $I_{\bar{\xi}} : \mathbb{R} \to \{0, 1\}$ is defined as

$$I_{\bar{\xi}}(V) = \begin{cases} 0 & \text{if } V \leq \bar{\xi} \\ 1 & \text{if } V > \bar{\xi}. \end{cases} \tag{8.10}$$

Note that $V_{\boldsymbol{\pi}}(\cdot)$ is a deterministic function, such that the expectation affects only the random variable $\mathbf{x}^+$ in (8.9). Moreover, $\beta$ is positive, $\exp(0) = 1$ and the exponential function is strictly increasing and positive. Therefore, we can bound the indicator function through the exponential expression

$$I_{\bar{\xi}}(V_{\boldsymbol{\pi}}(\mathbf{x}^+)) \leq \exp\left(\beta\left(V_{\boldsymbol{\pi}}(\mathbf{x}^+) - \bar{\xi}\right)\right) \tag{8.11}$$

due to the positivity of $\beta$. By taking the expectation of both sides, this inequality immediately

leads to

$$\mathbb{P}\left(V_\pi(\mathbf{x}^+) \geq \bar{\xi}\right) \leq \mathbb{E}_{\mathbf{x}^+}\left[\exp\left(\beta V_\pi(\mathbf{x}^+)\right)\right] \exp(-\beta\bar{\xi}). \tag{8.12}$$

Due to the definition of the risk operator in (8.6), we can simplify the right side of this inequality to obtain

$$\mathbb{P}\left(V_\pi(\mathbf{x}^+) \geq \bar{\xi}\right) \leq \exp\left(\beta\left(\mathbb{R}_\beta[V_\pi(\mathbf{x}^+)] - \bar{\xi}\right)\right). \tag{8.13}$$

Since $\mathbb{R}_\beta[V_\pi(\mathbf{x}^+)] \leq \xi$ is ensured by (8.7), we have $\mathbb{P}\left(V_\pi(\mathbf{x}^+) \geq \bar{\xi}\right) \leq \delta$ with $\delta$ defined in (8.8). □

This result provides a straightforward condition, which merely requires the evaluation of the risk operator and the computation of the cumulative cost, which is a problem commonly encountered in reinforcement learning. Moreover, it offers a simple expression for the probability of safety, such that it can easily be computed in practice.

**Remark 18.** *Since the probability of a safety violation $\delta$ guaranteed by proposition 3 only depends on three parameters, it allows an intuitive interpretation:*

- *The difference between $\xi$ and $\bar{\xi}$ can be interpreted as a safety margin since it requires the dynamics to be contractive on the set $\mathbb{V}_\pi^{\bar{\xi}} \setminus \mathbb{V}_\pi^{\xi}$ towards $\mathbb{V}_\pi^{\xi}$. The larger this safety margin, the more contractive is the behavior at the boundary of $\mathbb{V}_\pi^{\bar{\xi}}$ and consequently, it becomes more unlikely that the state reaches $\mathbb{X} \setminus \mathbb{V}_\pi^{\bar{\xi}}$.*

- *The parameter $\beta$ reflects the risk-sensitivity of the safety condition (8.7). A large value*

*of* $\beta$ *corresponds to a high risk-aversion since it causes the tails of the noise distribution* $\rho$ *and the function distribution* $\mathcal{F}$ *to have a larger effect on the left side of* (8.7)*. In the extreme case of* $\beta \to \infty$*, this leads to* (8.7) *corresponding to a condition on the worst case realization of* $\omega_k$ *and* $f(\cdot)$ *[149]. This increasing risk-aversion with growing* $\beta$ *is intuitively accompanied by an increase in the probability of safety.*

### 8.4.2 Safe backup Policies via Reinforcement Learning

While section 8.4.1 describes an approach for obtaining the probability of safety for a given policy, it does not address the problem of determining a safe policy. In this section, we show that this problem can be solved using standard reinforcement learning techniques through the following minimization problem

$$\pi_{\text{safe}} = \arg\min_{\pi \in \Pi} \mathbb{E}_x \left[ V_\pi(x) \right]. \tag{8.14}$$

Even though this optimization problem does not involve the risk operator $\mathbb{R}_\beta[\cdot]$, its solution $\pi_{\text{safe}}$ is guaranteed to satisfy the conditions of proposition 3 under weak assumptions. This is demonstrated by the subsequent theorem. The proof follows after a discussion of the assumptions.

**Theorem 13.** *Consider a cost function* $c(\cdot)$ *satisfying* (8.5) *and assume that there exist a policy* $\tilde{\pi}(\cdot)$ *and constants* $\theta_1, \theta_2 \in \mathbb{R}_+$ *with* $\theta_1 < 1/(1-\gamma)$ *such that*

$$V_\pi(x) \leq \theta_1 c(x) + \theta_2, \quad \forall x \in \mathbb{X} \tag{8.15}$$

*is satisfied. Moreover, assume there exist constants* $\theta_3, \theta_4 \in \mathbb{R}_{0,+}$ *such that*

$$V_\pi(\mathbf{x}) \geq \theta_3 c(\mathbf{x}) + \theta_4, \quad \forall \mathbf{x} \in \mathbb{X} \tag{8.16}$$

*holds for all policies* $\boldsymbol{\pi}(\cdot)$*. If*

$$\hat{c} > \frac{\theta_2}{\theta_3(\theta_1(\gamma - 1) + 1)} - \frac{\theta_4}{\theta_3} \tag{8.17}$$

*holds, then, the policy* (8.14) *is* $\delta^*$*-safe on* $\mathbb{V}_{\xi^*}$ *with* $\delta^* = \exp\left(\beta^*\left(\xi^* - \bar{\xi}\right)\right)$*, where*

$$\beta^*, \xi^* = \underset{\beta \in \mathbb{R}_+, \xi \in \mathbb{R}_+}{\arg\min} \exp\left(\beta\left(\xi - \bar{\xi}\right)\right) \tag{8.18a}$$

$$\text{s.t. } \xi < \bar{\xi} \tag{8.18b}$$

$$(8.7) \text{ holds.} \tag{8.18c}$$

Discussion While large values for $\theta_3$ and $\theta_4$ in (8.16) are generally beneficial for admitting larger values of $\hat{c}$ in (8.17), it is always possible to trivially choose $\theta_3 = 1$, $\theta_4 = 0$ due to non-negativity of $c(\cdot)$. Condition (8.15) essentially requires a sufficiently fast decay of the immediate costs $c(\mathbf{x}_k)$ along trajectories for some policy $\tilde{\boldsymbol{\pi}}(\cdot)$. This decay can be achieved if, e.g., variants of exponential controllability hold [150]. Since merely the existence of a policy $\tilde{\boldsymbol{\pi}}(\cdot)$ satisfying (8.15) is necessary, this admits the derivation of the constants $\theta_1$ and $\theta_2$ via properties such as exponential controllability [150]. Therefore, the assumptions of theorem 13 are not restrictive in practice.

Note that the required lower bound (8.16) for all possible cost functions $V_\pi(\cdot)$ is only

necessary because of the offset $\theta_2$, which leads to a lower bound for the admissible values of $\bar{\xi}$. Since the admissible value $\bar{\xi}$ depends directly on the cost function $V_\pi(\cdot)$, it indirectly depends on the policy $\pi(\cdot)$. Therefore, $V_{\tilde{\pi}}(\cdot)$ and $V_{\pi_{\text{safe}}}(\cdot)$ potentially admit different values for $\bar{\xi}$, such that general constraints cannot be posed on $\bar{\xi}$. This issue is resolved by (8.16), which establishes a direct relationship between $\hat{c}$ and $\bar{\xi}$ for all possible cost functions $V_\pi(\cdot)$ and thereby leads to the lower bound (8.17). If no offset exists, i.e., $\theta_2 = \theta_4 = 0$, it can be easily seen that $\hat{c} > 0$ must be satisfied. This is the trivial lower bound for $\hat{c}$ due to the assumed non-negativity of immediate cost functions $c(\cdot)$. Therefore, the offset $\theta_2$ is the only reason for the restriction of the admissible threshold $\hat{c}$.

Proof In order to prove theorem 13, we first show that a risk-neutral variant of condition (8.7) guarantees the existence of parameters $\xi$ and $\beta$ satisfying the requirements of proposition 3.

**Lemma 14.** *Assume that*

$$\mathbb{E}_{x^+}[V_\pi(x^+)] \leq \tilde{\xi}, \qquad \forall x \in \mathbb{V}_{\bar{\xi}} \tag{8.19}$$

*holds for some constant* $\tilde{\xi} < \bar{\xi}$. *Then, there exist constants* $\beta \in \mathbb{R}_+$ *and* $\xi < \bar{\xi}$ *such that* (8.7) *is satisfied.*

*Proof.* By the Taylor series expansion of the exponential function, we have

$$\mathbb{R}_\beta[V_\pi(x^+)] = \frac{1}{\beta}\log\left(1+\beta\mathbb{E}_{x^+}[V_\pi(x^+)]+\frac{\beta^2}{2}\mathbb{E}_{x^+}\left[V_\pi^2(x^+)\right]+\ldots\right). \tag{8.20}$$

From the premise of the lemma, it follows that

$$\mathbb{R}_\beta[V_\pi(\mathbf{x}^+)] \leq \frac{1}{\beta}\log\left(1+\beta\tilde{\xi}+\frac{\beta^2}{2}\mathbb{E}_{\mathbf{x}^+}\left[V_\pi^2(\mathbf{x}^+)\right]+\dots\right). \tag{8.21}$$

Since $\log(1+a) < a$ for $a \in \mathbb{R}_+$ and by noting the positivity of $V_\pi(\mathbf{x}^+)$ and the risk-aversion parameter $\beta$, we have

$$\mathbb{R}_\beta[V_\pi(\mathbf{x}^+)] < \tilde{\xi} + \beta\left(\frac{1}{2}\mathbb{E}_{\mathbf{x}^+}\left[V_\pi^2(\mathbf{x}^+)\right]+\dots\right). \tag{8.22}$$

Since the second summand can be brought arbitrarily close to 0 by choosing a sufficiently small $\beta$, there exists a $\beta$ such that the right side of (8.22) is smaller than $\bar{\xi}$, which concludes the proof. □

The key idea behind this result is that (8.7) converges to (8.19) for $\beta \to 0$. Therefore, it is sufficient to determine a policy $\boldsymbol{\pi}$, which satisfies the risk-neutral condition (8.19), for ensuring (8.7) with a suitably small value of $\beta \in \mathbb{R}_+$.

Although (8.19) is a risk-neutral condition, it exhibits an expectation with respect to the next state $\mathbf{x}^+$. Therefore, it does not directly enable the applicability of standard RL techniques and consequently, it does not coincide with the acquisition function considered in the definition of the safe policy (8.14). In order to overcome this issue, we exploit (8.15) to relate $\mathbb{E}_{\mathbf{x}^+}[V_\pi(\mathbf{x}^+)]$ to $V_\pi(\mathbf{x})$. This is achieved using the following lemma.

**Lemma 15.** *Assume that there exist* $\theta_1, \theta_2 \in \mathbb{R}_+$ *with* $\theta_1 < 1/(1-\gamma)$ *such that* (8.15) *is satisfied.*

*Then, it holds that*

$$\mathbb{E}_{x^+}[V_\pi(x^+)] - V_\pi(x) \leq \frac{\theta_1 - \theta_1\gamma - 1}{\theta_1\gamma} V_\pi(x) + \frac{\theta_2}{\gamma\theta_1}. \tag{8.23}$$

*Proof.* By solving Bellman's identity

$$V_\pi(x) = c(x) + \gamma \mathbb{E}_{x^+}\left[V_\pi(x')\right], \tag{8.24}$$

for $\mathbb{E}_{x^+}\left[V_\pi(x')\right]$, we can express $\Delta V_\pi(x) = \mathbb{E}_{x^+}[V_\pi(x^+)] - V_\pi(x)$ as

$$\Delta V_\pi(x) = \frac{1}{\gamma}(-c(x) + (1-\gamma)V_\pi(x)). \tag{8.25}$$

Due to (8.15), we have

$$c(x) \geq \frac{V_\pi(x) - \theta_2}{\theta_1}, \tag{8.26}$$

which allows us to bound (8.25) by

$$\Delta V_\pi(x) \leq \frac{1}{\gamma}\left(-\frac{V_\pi(x) - \theta_2}{\theta_1} + (1-\gamma)V_\pi(x)\right). \tag{8.27}$$

Rearranging the terms on the right side finally yields

$$\Delta V_\pi \leq \frac{\theta_1 - \theta_1\gamma - 1}{\theta_1\gamma} V_\pi(x) + \frac{\theta_2}{\gamma\theta_1}, \tag{8.28}$$

where $(\theta_1 - \theta_1\gamma - 1)/\theta_1\gamma$ is guaranteed to be negative since $\theta_1 < 1/(1-\gamma)$ is assumed. □

lemma 15 ensures that the minimization of $V_\pi(\mathbf{x})$ also reduces $\mathbb{E}_{\mathbf{x}^+}[V_\pi(\mathbf{x}^+)]$. This directly allows proving theorem 13 in combination with lemma 14 as shown in the following.

*Proof of theorem 13:* It is straightforward to see that optimizing with respect to the expectation over $\mathbf{x}$ yields identical policies $\boldsymbol{\pi}_{\text{safe}}(\cdot)$ as the point-wise optimum $\pi_{\mathbf{x}}(\mathbf{x}) = \arg\min_{\pi \in \Pi} V_\pi(\mathbf{x})$ for a given $\mathbf{x}$ and a continuous transition function $\mathbf{f}(\cdot,\cdot,\cdot)$. Due to optimality of $\boldsymbol{\pi}_{\mathbf{x}}(\cdot)$, we additionally have the inequality $V_{\mathbf{x}}(\mathbf{x}) \leq V_{\bar{\pi}}(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{X}$. Therefore, it follows from lemma 15 that

$$\mathbb{E}[V_{\boldsymbol{\pi}_{\text{safe}}}(\mathbf{x}^+)] \leq \frac{1}{\gamma}\left(1 - \frac{1}{\theta_1}\right) V_{\boldsymbol{\pi}_{\text{safe}}}(\mathbf{x}) + \frac{\theta_2}{\gamma\theta_1}. \tag{8.29}$$

Since the right side of (8.29) is linear in $V_{\boldsymbol{\pi}_{\text{safe}}}(\mathbf{x})$, the maximum inside $\mathbb{V}_{\bar{\xi}}$ is achieved for $V_{\boldsymbol{\pi}_{\text{safe}}}(\mathbf{x}) = \bar{\xi}$. Therefore, we obtain the inequality

$$\bar{\xi} > \frac{1}{\gamma}\left(1 - \frac{1}{\theta_1}\right) \bar{\xi} + \frac{\theta_2}{\gamma\theta_1} \tag{8.30}$$

since lemma 14 requires $\mathbb{E}[V_{\boldsymbol{\pi}_{\text{safe}}}(\mathbf{x}^+)] \leq \xi < \bar{\xi}$. Solving for $\bar{\xi}$ and noting that $\bar{\xi} = \theta_3\hat{c} + \theta_4$ due to (8.16) yields

$$\theta_3\hat{c} + \theta_4 > \frac{\theta_2}{\theta_1(\gamma - 1) + 1}. \tag{8.31}$$

□

It is straightforward to see that (8.17) guarantees the satisfaction of this inequality, such that lemma 14 and proposition 3 ensure that (8.18) is feasible and results in a probability $\delta^* < 1$.

**Algorithm 6** Safe RL using Risk-Sensitive Filters

/* Solve (8.32) */

**while** *optimization not converged* **do**

  Sample function $\hat{f}(\cdot) \sim \mathcal{F}$ Roll-out policy $\pi^*(\cdot)$ on $\hat{f}(\cdot)$ Update $\pi^*(\cdot)$ using gathered system data

/* Solve (8.14) */

**while** *optimization not converged* **do**

  Sample function $\hat{f}(\cdot) \sim \mathcal{F}$ Roll-out policy $\pi_{\text{safe}}(\cdot)$ on $\hat{f}(\cdot)$ Update $\pi_{\text{safe}}(\cdot)$ using gathered system data

/* Safe roll-out via online optimization (8.33) */

Apply $\pi^*_{\text{safe}}(\cdot)$ to unknown system $f(\cdot)$

This immediately implies $\delta^*$-safety of $\pi_{\text{safe}}(\cdot)$ and thereby concludes the proof.

### 8.4.3 Risk-Sensitive Inhibitory Control for Safe Reinforcement Learning

Based on the safe policy $\pi_{\text{safe}}(\cdot)$ obtained using (8.14), we propose a risk-sensitive inhibitory control strategy for enabling safe RL as outlined in Alg. algorithm 6. For this purpose, we first obtain an optimal, potentially unsafe policy by solving the optimization problem

$$\pi^* = \arg\max_{\pi \in \Pi} \mathbb{E}_{f,\omega,x_0} \left[ \sum_{k=0}^{\infty} \gamma^k r(x_k, \pi(x_k)) \right], \tag{8.32}$$

where $r : \mathbb{X} \times \mathbb{U} \to \mathbb{R}_{0,+}$ denotes a reward function and $x_k$ is defined through the iterative application of (8.1) with $x_0 = x$ and $u_k = \pi(x_k)$. This problem can be solved using standard off-policy reinforcement learning algorithms such as soft actor-critic reinforcement learning [151]. Afterward, a safe backup policy $\pi_{\text{safe}}(\cdot)$ is computed by solving (8.14), which can be straightforwardly achieved using standard off-policy reinforcement learning techniques. Finally, we apply

the policy to the true system (8.1). For this roll-out, we employ the risk-sensitive filter

$$\boldsymbol{\pi}^*_{\text{safe}}(\mathbf{x}) = \arg\min_{\mathbf{u}\in\mathbb{U}} \|\boldsymbol{\pi}^*(\mathbf{x}) - \mathbf{u}\| \tag{8.33a}$$

$$\text{s.t. } \mathbb{R}_\beta[V_{\boldsymbol{\pi}_{\text{safe}}}(\mathbf{f}(\mathbf{x},\mathbf{u},\boldsymbol{\omega}))] \leq \xi^* \tag{8.33b}$$

which makes use of the safe backup policy $\boldsymbol{\pi}_{\text{safe}}(\cdot)$ through the cost function $V_{\boldsymbol{\pi}_{\text{safe}}}$ and minimally adjusts the policy $\boldsymbol{\pi}^*(\cdot)$ such that the safety condition (8.7) is satisfied.

Due to the safety filter (8.33), the state constraints $\mathbb{X}_{\text{safe}}$ can straightforwardly be considered in Alg. algorithm 6. In fact, $\delta$-safety of $\boldsymbol{\pi}^*_{\text{safe}}(\cdot)$ is directly inherited from the safe backup policy $\boldsymbol{\pi}_{\text{safe}}(\cdot)$ as shown in the following theorem.

**Theorem 16.** *Consider a cost function* $\mathrm{c}(\cdot)$ *satisfying* (8.5) *and a threshold* $\hat{\mathrm{c}}$*, for which* (8.17) *holds. Moreover, assume that there exists a policy* $\tilde{\boldsymbol{\pi}}(\cdot)$ *satisfying* (8.15) *with* $\theta_1 < 1/(1-\gamma)$ *for all* $\mathbf{x} \in \mathbb{X}_{\text{safe}}$*. Then, the safety filtered policy* (8.33) *is* $\delta^*$*-safe on* $\mathbb{V}^{\xi^*}_{\boldsymbol{\pi}_{\text{safe}}}$ *with* $\delta^* = \exp\left(\beta^*\left(\xi^* - \bar{\xi}\right)\right)$*, where* $\beta^*$ *and* $\xi^*$ *are defined in* (8.18)*.*

*Proof.* Due to theorem 13, $\boldsymbol{\pi}_{\text{safe}}(\cdot)$ defined in (8.14) satisfies (8.33b). Thus, the optimization problem (8.33) is guaranteed to be feasible for all states $\mathbf{x} \in \mathbb{V}^{\xi^*}_{\boldsymbol{\pi}_{\text{safe}}}$ with the trivial solution $\mathbf{u} = \boldsymbol{\pi}_{\text{safe}}(\mathbf{x})$. Finally, $\delta^*$-safety directly follows from proposition 3. □

While this theorem employs the optimal parameters $\beta^*$ and $\xi^*$, it immediately follows from the proof of theorem 13 that for every value $\xi$ with $\xi^* \leq \xi < \bar{\xi}$, there exists a $\beta \in \mathbb{R}_+$ satisfying (8.18b). Therefore, $\delta$-safety on $\mathbb{V}_\xi \supset \mathbb{V}_{\xi^*}$ with $\delta > \delta^*$ can be straightforwardly ensured in practice by choosing a sufficiently large value $\xi < \bar{\xi}$ and a suitably small value $\beta \in \mathbb{R}_+$.

**Remark 19.** *When* $\beta$ *becomes larger, the control becomes more pessimistic, and therefore, the*

*probability of safety generally increases. However, there exists a critical value at which the safety constraint (8.33b) becomes infeasible for all $\xi < \bar{\xi}$. That is, the control becomes too phobic to act. This resembles a well-known behavior in risk-sensitive control and RL commonly referred to as neurotic breakdown [152].*

## 8.5 Simulations

In this section, we evaluate the proposed risk-sensitive inhibitory control approach, described in Alg. algorithm 6, using the popular Mujoco Half-Cheetah environment [153]. The Half-Cheetah is a planar model of a large, cat-like robot with 6 actuated joints. The main goal is to maximize the robot's walking velocity with the least control effort possible, which is encoded in the default reward function. We consider the default model parameters for the Cheetah robot, but assume a body mass perturbed by a Gaussian distributed random variable with 0 mean and standard deviation 0.1. In order to obtain a challenging safety conditions, we set optimality and safety in a direct conflict similar as in [143] by constraining the velocity to $v \leq v_{\text{crit}}$, $v_{\text{crit}} = 2$. As cost function for the computation of the safe policy (8.14), $c(\mathbf{x}) = v - \underline{v}$ is employed with threshold $\hat{c} = 2 - \underline{v}$, where $\underline{v} = 10$ denotes the considered minimum velocity of the Half-Cheetah robot. This cost function encourages the robot to run with a negative velocity, such that the distance to the safety threshold velocity $v_{\text{crit}}$ is maximized. Note that the subtraction of $\underline{v}$ is necessary to ensure the non-negativity of the cost $c(\cdot)$ assumed in our derivations, but it merely causes a constant off-set in the cumulative cost $V_\pi(\cdot)$.

The optimal and safe policies are obtained using the Soft-Actor Critic (SAC) algorithm [151] with 400 training iterations each with 1000 time steps and the hyper-parameters provided

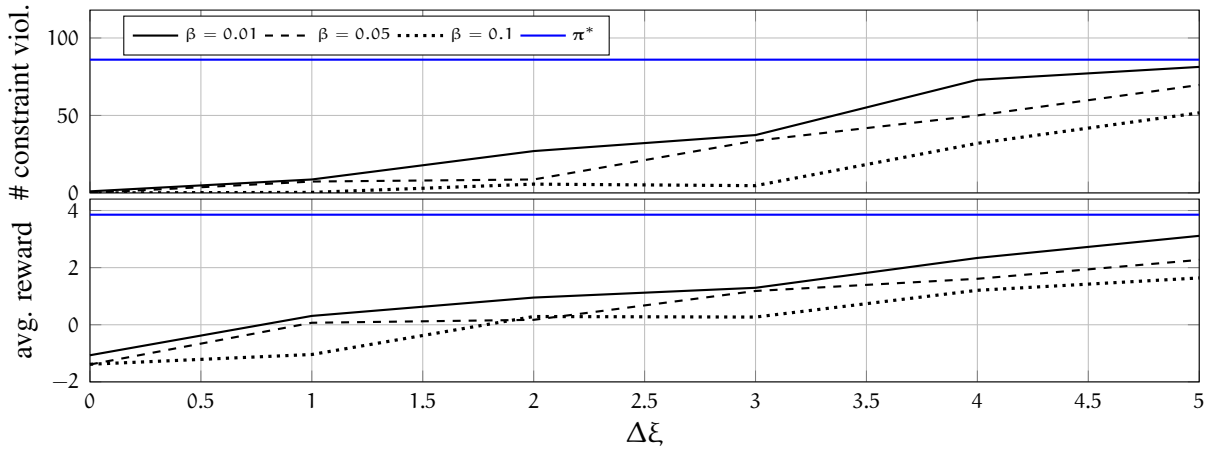



Figure 8.1: Number of constraint violations and average rewards in dependency on the safety constraint threshold $\xi = 521 + \Delta\xi$ and the risk-sensitivity $\beta$. Reducing $\beta$ and increasing $\xi$ have a similar effect of admitting more risky behavior in the response inhibition, such that the number of constraint violations and the average reward increase.

by [154]. For computing the expectations over dynamics $\mathbf{f}(\cdot)$ in (8.4) and (8.32), we randomly sample 10 body masses, such that we can use the corresponding sample environments to empirically approximate all necessary expected values. The risk-sensitive safety filter (8.33) is implemented using the cross-entropy method [155] with 5 iterations per time step and 10 particles. The safety constraints are considered in an augmented objective function using fixed Lagrange multipliers, such that they are effectively enforced using soft constraints to allow recovery after constraint violations. The risk operator $\mathbb{R}_\beta[\cdot]$ is approximated through 100 sample environments. For each parameter combination, $(\xi, \beta)$, 100 time steps are simulated and 3 random seeds are averaged.

The resulting numbers of constraint violations and the average reward for different values of $\beta$ and $\xi$ are depicted in fig. 8.1. We can observe that increasing $\xi$ has exactly the expected effect of loosening the safety constraint by admitting higher velocities $v$, such that the probability of safety decreases and more constraint violations can be observed. At the same time, this allows a higher robot velocity, which in turn causes an increasing average reward. A similar effect can

be observed with the risk parameter $\beta$ due to the considered state independent model uncertainty. When $\beta$ is increased, the conservatism of the safety filter increases. This leads to a lower number of constraint violations, but the average reward also reduces. Therefore, the parameters $\xi$ and $\beta$ exhibit the impact on the probability of safety as discussed in remark 18. Note that the risk-inhibition with the considered soft constraint formulation has a clearly visible effect on the average robot velocity, even when it does not manage to enforce the safety constraints. This can be observed in a comparison with the optimal policy $\boldsymbol{\pi}^*(\cdot)$, which achieves a significantly higher reward with a similar number of constraint violations for large values of $\xi$ and small $\beta$. Therefore, the proposed risk-sensitive inhibitory control not only allows to reduce the number of constraint violations, but also the amount by which the constraint is violated.

# Chapter 9: Conclusions and Future Work

Our theoretical results and experimental observations suggest that incorporating risk sensitivity is a practical approach to enhancing the robustness of both reinforcement learning (RL) algorithms and stochastic optimization problems. We first discuss a few directions that are worth further investigation. Then we make some concluding remarks.

1. **Alternative Risk-Measures.** Here, we focused our attention on exponential utility as the measure of risk, but a similar theoretic framework can be applied to a variety of risk measures, e.g., Value-at-Risk (VaR) – also referred to as the quantile – and Conditional VaR (CVaR).

2. **Alternative Algorithms.** In our study, we Incorporated risk sensitivity into two well-known algorithms: REINFORCE and Online Actor-Critic. However, the landscape of RL algorithms is diverse, and the concept of risk sensitivity can be extended to most, if not all, of them.

3. **Sample Complexity and Regret Analysis** Furthermore, our experimental observations demonstrated faster convergence and improved sample efficiency of risk-sensitive algorithms. These findings motivate a thorough analysis of sample complexity and regret for such algorithms.

4. **Experimental Validation.** While our current experiments have been confined to RL benchmarks, we recognize that their scope may not fully capture the potential advantages of our approach. Realistic experiments offer a broader perspective, allowing us to assess the performance of our methods in complex and dynamic environments that closely resemble real-world scenarios. By conducting experiments in more realistic settings, we can better evaluate the effectiveness and applicability of our approach, gaining insights into its performance, robustness, and adaptability in practical situations. This broader evaluation will provide a more comprehensive understanding of the advantages and limitations of our methods, enabling us to make informed decisions and recommendations for real-world applications.

5. **Multi-Agent Systems.** We took an empirical approach to the investigation of the role of risk-attitudes in the emergence of coordination in multi-agent environments in the absence of inter-agent communication and prior agreements. The simulation results confirmed our hypothesis that the agents' risk-attitudes shape the basin of attraction of the Nash equilibriums and with an appropriate choice of the risk parameters, the agents can efficiently converge to the Hicks optimal equilibrium. A theoretical analysis to characterize the dependence of the equilibriums' basin of attraction on the agents' risk attitudes, the complete analysis of N-agent systems in more complex games is the subject of our future work.

6. **Probabilistic Graphical Model Framework:** We have shown that risk-sensitive algorithms can be derived using the PGM formalism of RL. In doing so, we offered a systematic approach– a blueprint– for further algorithmic development in risk-sensitive RL. we explored the utility of EM algorithms for such problems. EM algorithms have well-known

advantages and disadvantages. For example, it is known that EM algorithms converge to the local optimum and have no guaranteed convergence rate. However, their simple and intuitive mechanism is the appeal of the EM algorithm as the first candidate and an effective method for developing risk-sensitive RL. In our future work, we will explore sampling-based EM algorithms that rely on step-wise samples. These algorithms may lead to RL algorithms with per-sample updates. We will also further explore the variants and alternatives to EM algorithms for developing risk-sensitive RL using the PGM modeling framework. Our main objective here was to provide a framework and a systematic approach to leverage PGM formalism for risk-sensitive RL and also illustrate the approach and results in simple examples.

# Bibliography


[1] Ronald A Howard. On Making Life and Death Decisions. In *Societal Risk Assessment*, pages 89–113. Springer, 1980.

[2] Tamer Basar and Pierre Bernhard. *H-infinity Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach*. Springer Science & Business Media, 2008.

[3] Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–291, 1979.

[4] Amos Tversky and Daniel Kahneman. Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and uncertainty*, 5(4):297–323, 1992.

[5] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level Control Through Deep Reinforcement Learning. *Nature*, 518(7540):529–533, 2015.

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

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

[8] Jan Leike, Miljan Martic, Victoria Krakovna, Pedro A Ortega, Tom Everitt, Andrew Lefrancq, Laurent Orseau, and Shane Legg. Ai safety gridworlds. *arXiv preprint arXiv:1711.09883*, 2017.

[9] Stuart Russell, Daniel Dewey, and Max Tegmark. Research priorities for robust and beneficial artificial intelligence. *Ai Magazine*, 36(4):105–114, 2015.

[10] Janosch Moos, Kay Hansel, Hany Abdulsamad, Svenja Stark, Debora Clever, and Jan Peters. Robust reinforcement learning: A review of foundations and recent advances. *Machine Learning and Knowledge Extraction*, 4(1):276–315, 2022.

[11] Takayuki Osogami. Robustness and risk-sensitivity in markov decision processes. *Advances in Neural Information Processing Systems*, 25, 2012.

[12] Erfaun Noorani and John S Baras. Risk-sensitive reinforce: A monte carlo policy gradient algorithm for exponential performance criteria. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 1522–1527. IEEE, 2021.

[13] LA Prashanth, Michael C Fu, et al. Risk-sensitive reinforcement learning via policy gradient search. *Foundations and Trends® in Machine Learning*, 15(5):537–693, 2022.

[14] Lars Peter Hansen and Thomas J Sargent. Robustness. In *Robustness*. Princeton university press, 2011.

[15] Herbert E Scarf. *A min-max solution of an inventory problem*. Rand Corporation Santa Monica, 1957.

[16] Lloyd S Shapley. Stochastic games. *Proceedings of the national academy of sciences*, 39(10):1095–1100, 1953.

[17] M. R. James and J. S. Baras. Robust $H_\infty$ output feedback control for nonlinear systems. *IEEE Transactions on Automatic Control*, 40(6):1007–1017, 1995.

[18] Santiago Paternain, Luiz Chamon, Miguel Calvo-Fullana, and Alejandro Ribeiro. Constrained reinforcement learning has zero duality gap. *Advances in Neural Information Processing Systems*, 32, 2019.

[19] Erick Delage and Shie Mannor. Percentile optimization for markov decision processes with parameter uncertainty. *Operations research*, 58(1):203–213, 2010.

[20] Mohammed Amin Abdullah, Hang Ren, Haitham Bou Ammar, Vladimir Milenkovic, Rui Luo, Mingtian Zhang, and Jun Wang. Wasserstein robust reinforcement learning, 2019.

[21] Aviv Tamar, Huan Xu, and Shie Mannor. Scaling up robust mdps by reinforcement learning, 2013.

[22] Xinlei Pan, Daniel Seita, Yang Gao, and John Canny. Risk averse robust adversarial reinforcement learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8522–8528. IEEE, 2019.

[23] Aviv Tamar Dotan Di Castro and Shie Mannor. Policy Gradients with Variance Related Risk Criteria. In *Proceedings of the 29th International Conference on Machine Learning, Edinburgh, Scotland, UK*, 2012.

[24] L. A. Prashanth. Policy gradients for cvar-constrained mdps. In Peter Auer, Alexander Clark, Thomas Zeugmann, and Sandra Zilles, editors, *Algorithmic Learning Theory*, pages 155–169, Cham, 2014. Springer International Publishing.

[25] Aviv Tamar. Risk-Sensitive and Efficient Reinforcement Learning Algorithms, 2015.

[26] Bo Liu, Ji Liu, and Kenan Xiao. R2PG: Risk-Sensitive and Reliable Policy Gradient. In *The Workshops of the The Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*, volume WS-18 of *AAAI Workshops*, pages 682–687. AAAI Press, 2018.

[27] D. Nass, B. Belousov, and J. Peters. Entropic Risk Measure in Policy Search. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1101–1106, 2019.

[28] Yinlam Chow and Mohammad Ghavamzadeh. Algorithms for CVaR Optimization in MDPs. *Advances in Neural Information Processing Systems*, 27:3509–3517, 2014.

[29] Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust Adversarial Reinforcement Learning. In *International Conference on Machine Learning*, pages 2817–2826. PMLR, 2017.

[30] Benjamin Eysenbach and Sergey Levine. Maximum entropy rl (provably) solves some robust rl problems, 2021.

[31] Emanuel Todorov. Linearly-solvable Markov Decision Problems. In *Advances in Neural Information Processing Systems*, pages 1369–1376, 2007.

[32] Alexandre Galashov, Siddhant M. Jayakumar, Leonard Hasenclever, Dhruva Tirumala, Jonathan Schwarz, Guillaume Desjardins, Wojciech M. Czarnecki, Yee Whye Teh, Razvan Pascanu, and Nicolas Manfred Otto Heess. Information Asymmetry in KL-regularized RL. *ArXiv*, abs/1905.01240, 2019.

[33] RONALD J. Williams and Jing Peng. Function Optimization Using Connectionist Reinforcement Learning Algorithms. *Connection Science*, 3(3):241–268, 1991.

[34] Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum Entropy Inverse Reinforcement Learning. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3*, AAAI'08, page 1433–1438. AAAI Press, 2008.

[35] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement Learning with Deep Energy-Based Policies. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1352–1361. JMLR.org, 2017.

[36] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1861–1870, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[37] Yingjie Fei, Zhuoran Yang, Yudong Chen, Zhaoran Wang, and Qiaomin Xie. Risk-sensitive reinforcement learning: Near-optimal risk-sample tradeoff in regret. *Advances in Neural Information Processing Systems*, 33:22384–22395, 2020.

[38] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, 2017.

[39] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust Region Policy Optimization. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1889–1897, Lille, France, 07–09 Jul 2015. PMLR.

[40] Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation, 2018.

[41] Vivek S Borkar. Learning algorithms for risk-sensitive control. *Proceedings of the 19th International Symposium on Mathematical Theory of Networks and Systems*, 5(9), 2010.

[42] Erfaun Noorani and John S Baras. Risk-sensitive reinforcement learning and robust learning for control. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 2976–2981. IEEE, 2021.

[43] David Jacobson. Optimal Stochastic Linear Systems with Exponential Performance Criteria and Their Relation to Deterministic Differential Games. *IEEE Transactions on Automatic Control*, 18(2):124–131, 1973.

[44] Erfaun Noorani and John S Baras. Embracing risk in reinforcement learning: The connection between risk-sensitive exponential and distributionally robust criteria. In *2022 American Control Conference (ACC)*, pages 2703–2708. IEEE, 2022.

[45] Martin L Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

[46] Erfaun Noorani, Christos N Mavridis, and John S Baras. Exponential td learning: A risk-sensitive actor-critic reinforcement learning algorithm. In *2023 American Control Conference (ACC)*, pages 2697–2702. IEEE, 2023.

[47] Erfaun Noorani, Christos Mavridis, and John Baras. Risk-sensitive reinforcement learning with exponential criteria. *arXiv preprint arXiv:2212.09010*, 2022.

[48] Erfaun Noorani and John S Baras. A probabilistic perspective on risk-sensitive reinforcement learning. In *2022 American Control Conference (ACC)*, pages 2697–2702. IEEE, 2022.

[49] Hans Föllmer and Alexander Schied. Convex Measures of Risk and Trading Constraints. *Finance and stochastics*, 6(4):429–447, 2002.

[50] Marco Frittelli and Emanuela Rosazza Gianin. Putting Order in Risk Measures. *Journal of Banking & Finance*, 26(7):1473–1486, 2002.

[51] Ronald A Howard and James E Matheson. Risk-sensitive markov decision processes. *Management science*, 18(7):356–369, 1972.

[52] Jason Speyer, John Deyst, and D Jacobson. Optimization of Stochastic Linear Systems With Additive Measurement and Process Noise Using Exponential Performance Criteria. *IEEE Transactions on Automatic Control*, 19(4):358–366, 1974.

[53] PR Kumar and JH Van Schuppen. On The Optimal Control of Stochastic Systems With an Exponential-of-integral Performance Index. *Journal of mathematical analysis and applications*, 80(2):312–332, 1981.

[54] Matthew R James, John S Baras, and Robert J Elliott. Risk-sensitive Control and Dynamic Games for Partially Observed Discrete-time Nonlinear Systems. *IEEE transactions on automatic control*, 39(4):780–792, 1994.

[55] J. S. Baras and M. R. James . Robust and Risk-sensitive Output Feedback Control for Finite State Machines and Hidden Markov Models. *Journal of Mathematical Systems, Estimation, and Control*, 7(3):371–374, 1997.

[56] Matthew R James and JS Baras. Partially Observed Differential Games, Infinite-Dimensional Hamilton–Jacobi–Isaacs Equations, and Nonlinear $H_\infty$ Control. *SIAM Journal on Control and Optimization*, 34(4):1342–1364, 1996.

[57] J. S. Baras and N. S. Patel. Robust Control of Set-valued Discrete-time Dynamical Systems. *IEEE Transactions on Automatic Control*, 43(1):61–75, 1998.

[58] SR Srinivasa Varadhan. *Large deviations and applications*. SIAM, 1984.

[59] A Dembod, O Zeltouni, and K Fleischmann. Large Deviations Techniques and Applications. *Jahresbericht der Deutschen Mathematiker Vereinigung*, 98(3):18–18, 1996.

[60] Hugo Touchette. The Large Deviation Approach to Statistical Mechanics. *Physics Reports*, 478(1-3):1–69, 2009.

[61] Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. Coherent measures of risk. *Mathematical finance*, 9(3):203–228, 1999.

[62] Liyiming Ke, Matt Barnes, Wen Sun, Gilwoo Lee, Sanjiban Choudhury, and Siddhartha Srinivasa. Imitation Learning as f-Divergence Minimization. *arXiv preprint arXiv:1905.12888*, 2019.

[63] Lex Weaver and Nigel Tao. The optimal reward baseline for gradient-based reinforcement learning, 2013.

[64] Christos Mavridis, Erfaun Noorani, and John S Baras. Risk sensitivity and entropy regularization in prototype-based learning. In *2022 30th Mediterranean Conference on Control and Automation (MED)*, pages 194–199. IEEE, 2022.

[65] Paolo Dai Pra, Lorenzo Meneghini, and Wolfgang J Runggaldier. Connections between stochastic control and dynamic games. *Mathematics of Control, Signals and Systems*, 9(4):303–326, 1996.

[66] Harry Markowitz. Portfolio Selection. *The Journal of Finance*, 7(1):77–91, 1952.

[67] Keith Glover and John C Doyle. State-space Formulae for All Stabilizing Controllers That Satisfy An H-$\infty$-norm Bound and Relations to Risk Sensitivity. *Systems & control letters*, 11(3):167–172, 1988.

[68] Amir Ahmadi-Javid. Entropic Value-at-Risk: A New Coherent Risk Measure. *Journal of Optimization Theory and Applications*, 155(3):1105–1123, 2012.

[69] Hans Föllmer and Thomas Knispel. Entropic Risk Measures: Coherence vs. Convexity, Model Ambiguity and Robust Large Deviations. *Stochastics and Dynamics*, 11(02n03):333–351, 2011.

[70] James C. Bezdek and Richard J. Hathaway. Some notes on alternating optimization. In Nikhil R. Pal and Michio Sugeno, editors, *Advances in Soft Computing — AFSS 2002*, pages 288–300, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[71] H. J. Kappen and H. C. Ruiz. Adaptive Importance Sampling for Control and Inference. *Journal of Statistical Physics*, 162(5):1244–1266, 2016.

[72] Meng Xu and José M. Angulo. Divergence-based risk measures: A discussion on sensitivities and extensions. *Entropy*, 21(7):1–18, 2019.

[73] Alfréd Rényi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pages 547–561. University of California Press, 1961.

[74] Syed Mumtaz Ali and Samuel D Silvey. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society: Series B (Methodological)*, 28(1):131–142, 1966.

[75] Imre Csiszár. Eine informationstheoretische ungleichung und ihre anwendung auf beweis der ergodizitaet von markoffschen ketten. *Magyer Tud. Akad. Mat. Kutato Int. Koezl.*, 8:85–108, 1964.

[76] Tetsuzo Morimoto. Markov processes and the h-theorem. *Journal of the Physical Society of Japan*, 18(3):328–331, 1963.

[77] Boris Belousov and Jan Peters. f-divergence constrained policy improvement. *arXiv preprint arXiv:1801.00056*, 2017.

[78] 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, 2020.

[79] J. N. Tsitsiklis and B. Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5):674–690, 1997.

[80] Frank L Lewis, Draguna Vrabie, and Kyriakos G Vamvoudakis. Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers. *IEEE Control Systems Magazine*, 32(6):76–105, 2012.

[81] Shixiang Gu, Timothy Lillicrap, Ilya Sutskever, and Sergey Levine. Continuous deep q-learning with model-based acceleration. In *International Conference on Machine Learning*, pages 2829–2838. PMLR, 2016.

[82] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy Gradient Methods for Reinforcement Learning With Function Approximation. In *Advances in Neural Information Processing Systems*, pages 1057–1063, 2000.

[83] Sham M Kakade. A Natural Policy Gradient. *Advances in Neural Information Processing Systems*, 14:1531–1538, 2001.

[84] Shixiang Shane Gu, Timothy Lillicrap, Richard E Turner, Zoubin Ghahramani, Bernhard Schölkopf, and Sergey Levine. Interpolated Policy Gradient: Merging On-policy and Off-policy Gradient Estimation for Deep Reinforcement Learning. In *Advances in Neural Information Processing Systems*, pages 3846–3855, 2017.

[85] Vijay Konda and John Tsitsiklis. Actor-critic algorithms. *Advances in neural information processing systems*, 12, 1999.

[86] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

[87] Philip S Thomas and Emma Brunskill. Policy Gradient Methods for Reinforcement Learning with Function Approximation and Action-dependent Baselines, 2017.

[88] Will Grathwohl, Dami Choi, Yuhuai Wu, Geoffrey Roeder, and David Duvenaud. Backpropagation Through The Void: Optimizing Control Variates for Black-box Gradient Estimation, 2017.

[89] Hao Liu, Yihao Feng, Yi Mao, Dengyong Zhou, Jian Peng, and Qiang Liu. Action-depedent Control Variates for Policy Optimization via Stein's Identity, 2017.

[90] Cathy Wu, Aravind Rajeswaran, Yan Duan, Vikash Kumar, Alexandre M Bayen, Sham Kakade, Igor Mordatch, and Pieter Abbeel. Variance Reduction for Policy Gradient with Action-dependent Factorized Baselines, 2018.

[91] George Tucker, Surya Bhupatiraju, Shixiang Gu, Richard Turner, Zoubin Ghahramani, and Sergey Levine. The Mirage of Action-Dependent Baselines in Reinforcement Learning. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5015–5024, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[92] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

[93] Vivek S Borkar. *Stochastic approximation: a dynamical systems viewpoint*, volume 48. Springer, 2009.

[94] Prashanth La and Mohammad Ghavamzadeh. Actor-critic algorithms for risk-sensitive mdps. *Advances in neural information processing systems*, 26, 2013.

[95] Vijay R. Konda and John N. Tsitsiklis. On Actor-Critic Algorithms. *SIAM J. Control Optim.*, 42(4):1143–1166, April 2003.

[96] Shalabh Bhatnagar, Richard S. Sutton, Mohammad Ghavamzadeh, and Mark Lee. Natural actor–critic algorithms. *Automatica*, 45(11):2471–2482, 2009.

[97] Christos N Mavridis and John S Baras. Vector quantization for adaptive state aggregation in reinforcement learning. In *2021 American Control Conference (ACC)*, pages 2187–2192. IEEE, 2021.

[98] Christos N Mavridis, Nilesh Suriyarachchi, and John S Baras. Maximum-entropy progressive state aggregation for reinforcement learning. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 5144–5149. IEEE, 2021.

[99] Christos N. Mavridis and John S. Baras. Annealing optimization for progressive learning with stochastic approximation. *IEEE Transactions on Automatic Control*, 70:1–13, 2022.

[100] Vivek S Borkar. Q-learning for risk-sensitive control. *Mathematics of operations research*, 27(2):294–311, 2002.

[101] Yingjie Fei, Zhuoran Yang, Yudong Chen, and Zhaoran Wang. Exponential bellman equation and improved regret bounds for risk-sensitive reinforcement learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[102] Andrew G Barto, Richard S Sutton, and Charles W Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE transactions on systems, man, and cybernetics*, SMC-13(5):834–846, 1983.

[103] Richard S Sutton. Generalization in reinforcement learning: Successful examples using sparse coarse coding. *Advances in neural information processing systems*, 8, 1995.

[104] Radford M Neal and Geoffrey E Hinton. A View of The EM Algorithm that Justifies Incremental, Sparse, and Other variants. In *Learning in graphical models*, pages 355–368. Springer, 1998.

[105] Frank Dellaert. The Expectation Maximization Algorithm. Technical report, Georgia Institute of Technology, 2002.

[106] Lei Xu and Michael I Jordan. On Convergence Properties of the EM Algorithm for Gaussian Mixtures. *Neural computation*, 8(1):129–151, 1996.

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

[108] Jens Kober and Jan Peters. Policy Search for Motor Primitives in Robotics. *Advances in Neural Information Processing Systems*, 21, 2008.

[109] Marc Peter Deisenroth, Gerhard Neumann, Jan Peters, et al. A Survey on Policy Search for Robotics. *Foundations and trends in Robotics*, 2(1-2):388–403, 2013.

[110] Xitong Yang. Understanding The Variational Lower Bound, 2017.

[111] Yunlong Song and Davide Scaramuzza. Policy Search for Model Predictive Control With Application to Agile Drone Flight. *IEEE Transactions on Robotics*, 2022.

[112] Kenneth J Arrow. *The Limits of Organization*. WW Norton & Company, 1974.

[113] Mürüvvet Büyükboyacı. Risk Attitudes and The Stag-Hunt Game. *Economics Letters*, 124(3):323–325, 2014.

[114] Erfaun Noorani and John S Baras. Risk-sensitive reinforce: A monte carlo policy gradient algorithm for exponential performance criteria. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 1522–1527. IEEE, 2021.

[115] John C Harsanyi, Reinhard Selten, et al. A General Theory of Equilibrium Selection in Games. *MIT Press Books*, 1, 1988.

[116] Werner Güth. Equilibrium Selection by Unilateral Deviation Stability. In *Rational interaction*, pages 161–189. Springer, 1992.

[117] Jean-Jacques Rousseau. *A Discourse on Inequality*. Penguin, 1984.

[118] Robert Axelrod and William Donald Hamilton. The Evolution of Cooperation. *science*, 211(4489):1390–1396, 1981.

[119] Ming Tan. Multi-agent Reinforcement Learning: Independent vs. Cooperative Agents. In *Proceedings of the tenth international conference on machine learning*, pages 330–337, 1993.

[120] Taiki Fuji, Kiyoto Ito, Kohsei Matsumoto, and Kazuo Yano. Deep multi-agent Reinforcement Learning Using DNN-weight Evolution to Optimize Supply Chain Performance. In *Proceedings of the 51st Hawaii International Conference on System Sciences*, 2018.

[121] Ardi Tampuu, Tambet Matiisen, Dorian Kodelja, Ilya Kuzovkin, Kristjan Korjus, Juhan Aru, Jaan Aru, and Raul Vicente. Multiagent Cooperation and Competition with Deep Reinforcement Learning. *PloS one*, 12(4), 2017.

[122] Jakob Foerster, Nantas Nardelli, Gregory Farquhar, Triantafyllos Afouras, Philip HS Torr, Pushmeet Kohli, and Shimon Whiteson. Stabilising Experience Replay for Deep Multi-agent Reinforcement Learning. In *International conference on machine learning*, pages 1146–1155. PMLR, 2017.

[123] Yu Bai and Chi Jin. Provable Self-play Algorithms for Competitive Reinforcement Learning. In *International Conference on Machine Learning*, pages 551–560. PMLR, 2020.

[124] Jakob N Foerster, Richard Y Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with Opponent-learning Awareness. *arXiv preprint arXiv:1709.04326*, 2017.

[125] Jiequn Han and Ruimeng Hu. Deep Fictitious Play for Finding Markovian Nash Equilibrium in Multi-agent Games. In *Mathematical and Scientific Machine Learning*, pages 221–245. PMLR, 2020.

[126] Johannes Heinrich and David Silver. Deep Reinforcement Learning from Self-play in Imperfect-Information Games. *arXiv preprint arXiv:1603.01121*, 2016.

[127] Nitin Kamra, Umang Gupta, Kai Wang, Fei Fang, Yan Liu, and Milind Tambe. Deep Fictitious Play for Games with Continuous Action Spaces. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 2042–2044, 2019.

[128] Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. A Unified Game-theoretic Approach to Multiagent Reinforcement Learning. *arXiv preprint arXiv:1711.00832*, 2017.

[129] Julien Perolat, Remi Munos, Jean-Baptiste Lespiau, Shayegan Omidshafiei, Mark Rowland, Pedro Ortega, Neil Burch, Thomas Anthony, David Balduzzi, Bart De Vylder, et al. From Poincaré Recurrence to Convergence in Imperfect Information Games: Finding Equilibrium via Regularization. In *International Conference on Machine Learning*, pages 8525–8535. PMLR, 2021.

[130] Michael L Littman. Markov Games as a Framework for Multi-agent Reinforcement Learning. In *Machine learning proceedings 1994*, pages 157–163. Elsevier, 1994.

[131] Zhenggang Tang, Chao Yu, Boyuan Chen, Huazhe Xu, Xiaolong Wang, Fei Fang, Simon Du, Yu Wang, and Yi Wu. Discovering Diverse Multi-Agent Strategic Behavior via Reward Randomization. *arXiv preprint arXiv:2103.04564*, 2021.

[132] Alexander Peysakhovich and Adam Lerer. Prosocial learning agents solve generalized stag hunts better than selfish ones. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, page 2043–2044, Richland, SC, 2018. International Foundation for Autonomous Agents and Multiagent Systems.

[133] Armin Lederer, Erfaun Noorani, Sandra Hirche, and John S Baras. Risk-sensitive reinforce: A monte carlo policy gradient algorithm for exponential performance criteria. In *2023 62nd IEEE Conference on Decision and Control (CDC)*. IEEE, 2023.

[134] Joel T. Nigg. On Inhibition/Disinhibition in Developmental Psychopathology: Views from Cognitive and Personality Psychology and a Working Inhibition Taxonomy. *Psychological Bulletin*, 126(2):220–246, 2000.

[135] Lukas Brunke, Melissa Greeff, Adam W. Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P. Schoellig. Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5:411–444, 2022.

[136] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2017.

[137] Gabriel Dulac-Arnold, Daniel Mankowitz, and Todd Hester. Challenges of Real-World Reinforcement Learning. In *ICML Workshop on Real-Life Reinforcement Learning*, 2019.

[138] Mohammad Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe Reinforcement Learning via Shielding. In *AAAI Conference on Artificial Intelligence*, pages 2669–2678, 2018.

[139] Andrew Taylor, Andrew Singletary, Yisong Yue, and Aaron Ames. Learning for Safety-Critical Control with Control Barrier Functions. In *Learning for Dynamics & Control*, pages 708–717, 2019.

[140] Osbert Bastani. Safe Reinforcement Learning with Nonlinear Dynamics via Model Predictive Shielding. In *American Control Conference*, pages 3488–3494, 2021.

[141] Kim P. Wabersich, Lukas Hewing, Andrea Carron, and Melanie N. Zeilinger. Probabilistic Model Predictive Safety Certification for Learning-Based Control. *IEEE Transactions on Automatic Control*, 76(1):176–188, 2021.

[142] Kai Chieh Hsu, Vicenç Rubies-Royo, Claire J. Tomlin, and Jaime F. Fisac. Safety and Liveness Guarantees through Reach-Avoid Reinforcement Learning. In *Robotics: Science and Systems*, 2021.

[143] Sebastian Curi, Armin Lederer, Sandra Hirche, and Andreas Krause. Safe Reinforcement Learning via Confidence-Based Filters. In *IEEE Conference on Decision and Control*, 2022.

[144] Lauren Sherman, Laurence Steinberg, and Jason Chein. Connecting Brain Responsivity and Real-World Risk Taking: Strengths and Limitations of Current Methodological Approaches. *Developmental Cognitive Neuroscience*, 33:27–41, 2018.

[145] Mohamadreza Ahmadi, Xiaobin Xiong, and Aaron D. Ames. Risk-Averse Control via CVaR Barrier Functions: Application to Bipedal Robot Locomotion. *IEEE Control Systems Letters*, 6:878–883, 2022.

[146] Andrew Singletary, Mohamadreza Ahmadi, and Aaron D. Ames. Safe Control for Nonlinear Systems With Stochastic Uncertainty via Risk Control Barrier Functions. *IEEE Control Systems Letters*, 7:349–354, 2023.

[147] Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, Cambridge, MA, 2006.

[148] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *Advances in Neural Information Processing Systems*, pages 6405–6416, 2017.

[149] M.R. James, J.S. Baras, and R.J. Elliott. Risk-Sensitive Control and Dynamic Games for Partially Observed Discrete-Time Nonlinear Systems. *IEEE Transactions on Automatic Control*, 39(4):780–792, 1994.

[150] Vladimir Gaitsgory, Lars Grüne, Matthias Höger, Christopher M. Kellett, and Steven R. Weller. Stabilization of Strictly Dissipative Discrete Time Systems with Discounted Optimal Control. *Automatica*, 93:311–320, 2018.

[151] 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, 2018.

[152] Wendell H Fleming. Risk Sensitive Stochastic Control and Differential Games. *Communications in Information and Systems*, 6(3):161–177, 2006.

[153] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012.

[154] Eric Liang, Richard Liaw, Philipp Moritz, Robert Nishihara, Roy Fox, Ken Goldberg, Joseph E. Gonzalez, Michael I. Jordan, and Ion Stoica. RLlib: Abstractions for Distributed Reinforcement Learning. In *International Conference on Machine Learning*, pages 4768–4780, 2018.

[155] Zdravko I Botev, Dirk P Kroese, Reuven Y Rubinstein, and Pierre L'Ecuyer. The cross-entropy method for optimization. In *Handbook of Statistics*, volume 31, pages 35–59. Elsevier, 2013.