

Robust Decision-Making under Uncertainty: Data-Driven Uncertainty and Ambiguity Set Construction for Robust Optimisation and Robust Markov Decision Processes

Andreas Neofytou

Department of Management Science

University of Strathclyde, Glasgow

October 14, 2026

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

The copyright of this thesis belongs to the author under the terms of the United Kingdom Copyright Acts as qualified by University of Strathclyde Regulation 3.50. Due acknowledgement must always be made of the use of any material contained in, or derived from, this thesis.

Signed: Andreas Neofytou

Date: October 2026

Abstract

In today's dynamic and uncertain environments, Robust Decision-Making under Uncertainty is essential across diverse applications. However, conventional Robust Optimisation and Robust Markov Decision Processes suffer from overly conservative solutions. This is mainly due to simplistic or inaccurate uncertainty modelling. The heart of these fields is the proper construction of uncertainty and ambiguity sets that influence the outcome. In this thesis, we aim to explore these challenges by introducing novel, data-driven and machine learning approaches for constructing compact uncertainty sets. Through this, we seek to obtain significantly enhanced solution quality.

Specifically, the thesis presents two methodological innovations. First, we study a novel uncertainty modelling approach for Robust Optimisation, utilizing a Bootstrapped Dirichlet Process Mixture Model to generate compact data-driven uncertainty sets that capture complex uncertainty distributions effectively. Second, we introduce an innovative framework for Robust Markov Decision Processes by integrating Bootstrapped Deep Q-Network ensembles with Bayesian ambiguity set construction. Consequently, we dynamically estimate both the centre and size of ambiguity sets from data in an online manner.

Through extensive numerical experiments, the proposed methods demonstrate substantial improvements in solution quality, specifically in objective value for Robust Optimisation and worst-case return in Robust Markov Decision Processes, while outperforming conventional robust frameworks. We essentially addressed the fundamental challenge of overly conservative uncertainty modelling in robust decision-making by employing advanced statistical machine learning methods. Collectively, these contributions bridge predictive machine learning and prescriptive decision-making by translat-

ing predictive uncertainty into compact uncertainty sets that directly guide the Robust Optimisation and Robust Markov Decision Processes.

Contents

Abstract ii

List of Figures vi

List of Tables x

List of Notations xiii

List of Abbreviations and Acronyms xvii

Glossary of Key Terms xix

Acknowledgements xxi

Publications xxii

1 Introduction 1

1.1 Background and Motivation 1

1.2 Decision-making process 3

1.3 Uncertainty: Types and Modelling 4

1.3.1 Uncertainty Sets in Robust Optimisation 5

1.3.2 Ambiguity Sets in Robust Markov Decision Processes 6

1.4 Main Contributions 8

1.5 Thesis Structure Overview 11

2 Literature Review 12

2.1 Robust Optimisation 12

2.1.1 Classical robust optimisation and uncertainty sets 12
2.1.2 Machine Learning in the data-driven uncertainty set construction for robust optimisation 15
2.2 Robust Markov Decision Processes and Reinforcement Learning 19
2.2.1 Related studies in Robust Markov Decision Processes 19
2.2.2 Related studies in Exploration 21
2.3 Research Gaps and Opportunities 23

3 Compact Data-Driven Uncertainty Sets for Robust Optimisation 25
3.1 Introduction 25
3.2 Problem Statement 28
3.3 Methodology 32
3.3.1 Dirichlet process Gaussian mixture model to describe the data pattern 32
3.3.2 Construction of the data-driven uncertainty set 39
3.3.3 Probabilistic guarantees of the proposed method 49
3.4 Experiments 52
3.4.1 Motivating example – a toy problem 52
3.4.2 Production planning example 67
3.4.3 Robust vehicle pre-allocation example 71
3.4.4 Adaptive robust lot-sizing example 74
3.5 Concluding remarks 76

4 Compact Bayesian Ambiguity Sets for Robust Markov Decision Processes 78
4.1 Introduction 78
4.2 Problem Statement 82
4.3 Methodology 86
4.3.1 Step 1: Bootstrapped Q-Value exploration 87
4.3.2 Step 2: Bayesian Ambiguity Set construction 91
4.3.3 Step 3: Solving the RMDP 94

4.3.4 Theoretical properties 95
4.4 Experiments 97
4.4.1 RiverSwim domain 98
4.4.2 Inventory optimisation domain 108
4.5 Concluding remarks 114

5 Conclusions **116**
5.1 Research Findings 116
5.2 Research Limitations 121
5.3 Future Research 123

Bibliography **124**

List of Figures

1.1 Decision-Making under uncertainty framework 4

3.1 Fitting Gaussian Mixture Model(s) to find the true representation of the data. (a) Attempting to fit a Normal Distribution to represent the data, (b) Approximate the true posterior distribution representation 34

3.2 The process for Variational Inference in DPMM 40

3.3 Intersection of l_1 and l_∞ norms 42

3.4 Calculation of Gamma for l_∞ norm 43

3.5 Bootstrapping method with 1000 iteration resampling for a single cluster. How to calculate the means of bootstrapped μ for one single cluster and the μ_k^* of that k cluster. Comparison is also made with DPMM mean without bootstrapped resampling. 47

3.6 SCENARIO 1: Process for the construction of the Data-Driven Uncertainty Set. (a) initial data, (b) Outlier identification, (c) Inference in black the parameter mean of each predicted cluster 55

3.7 (a) and (b) are traditional Box and Budget Uncertainty Sets. (c) is our approach with 1 predefined component, (d) our approach with 3 predefined components. (e) is our approach with just l_∞ norm uncertainty set and (f) with just l_1 norm uncertainty set. 56

3.8 Proposed Bootstrap DPMM approach to construct the optimal number of components automatically 58

3.9 k-means algorithm, resulting uncertainty set construction and comparison 59

3.10 DBSCAN algorithm, resulting uncertainty set construction and comparison 60

List of Figures

3.11 Coverage Rate Comparisons over different percentages of the same data 61
3.12 Bootstrap vs Non-Bootstrap approach: Difference between uncertainty set size, solutions and objectives 62
3.13 SCENARIO 2: Our framework works well on correlated dataset with linear pattern. 62
3.14 SCENARIO 3: Asymmetric dataset variation spread with noise in irregular intervals. (a) Traditional Box Uncertainty Set, (b) Traditional Budget Uncertainty Set, (c) Clustering behaviour of k-means after its own outlier detection 63
3.15 SCENARIO 3: Proposed data-driven uncertainty sets on dataset with data variation 64
3.16 Train-test split and cross validation. Testing data fall inside the trained method and uncertainty set. 65
3.17 Testing on SCENARIO 1 to see how our approach behaves with additional 100 unseen out-of-sample data points over time: (a) Results after 15 new data points, (b) Results after 35 new data points, (c) Results after 100 new data points 66
3.18 Traditional Uncertainty Sets for Production Planning experiment. Box vs Budget Uncertainty Sets 69
3.19 Proposed data-driven approach for experiment 2 over different time periods (t=12) 71
3.20 Experiment 3: Allocation Quantity by region and comparison between Box vs epsilon-neighborhood vs proposed data-driven uncertainty sets . 73
3.21 Experiment 4: Lot sizing allocation. Difference allocation between (a) Budget uncertainty set and (b) proposed bootstrapped DPMM uncertainty set. 74

List of Figures

4.1 Bootstrapped DQN framework: Assuming $s = 6$ states as one-hot encoded input, and a shared hidden layer with 32 units and ReLU activation. The output involves K independent heads, each producing Q-values for $a = 2$ different actions. Initially, a single head is selected at random per episode and only the parameters of that specific head and the shared layers are updated, while the other heads remain frozen. The dashed black lines show the output heads that are active under the bootstrapped mechanism, for instance only Head 2 and Head K. 89
4.2 Flowchart of our framework 94
4.3 Exploration for RiverSwim: (a) Mean reward for each exploration method and (b) Cumulative reward for each exploration method 100
4.4 RiverSwim results over 20 independent runs, showing means and 95% Student-t confidence intervals as shaded regions in (a)–(c) and error bars in (d)–(f). Specifically: (a) the robust certificate A, (b) the true-model value B of the learned robust policy, (c) its evolution as additional exploration data are collected, (d) its final value for each method. Lower values indicate that the robust certificate is closer to the oracle return, (e) the absolute certificate gap $|B-A|$, and (f) the absolute policy-oracle gap $|C - B|$. Higher values are preferable in (a)-(b), while lower values are preferable in (c)-(f). 104
4.5 Policy schedule for RiverSwim after 750 training episodes: (a) policy for a single seed and (b) policy consistency with percentage of success for taking action 1, across 20 seeds. 105
4.6 RiverSwim sensitivity of the robust certificate $A = \min_{P \in \mathcal{P}} V_P^{\widehat{\pi}}(s_0)$ to the number of Bootstrapped DQN heads K after 750 exploration episodes. Markers show the means over 10 independent runs, error bars show 95% Student-t confidence intervals, and the dashed line indicates the previous oracle value $C = 959.44$. 107
4.7 Exploration for Inventory problem: (a) Mean reward for each exploration method and (b) Cumulative reward for each exploration method 110

4.8 Inventory optimisation: results over 10 independent runs, showing means and 95% Student-t confidence intervals as shaded regions in (a) and (c), while error bars in (b),(d)–(f). Specifically: (a) the robust certificate A, (b) the final true-model value B of the learned robust policy, (c) its evolution of error as additional exploration data are collected, (d) its final value for each method. Lower values indicate that the robust certificate is closer to the oracle return, (e) the absolute certificate gap $|B - A|$, and (f) the absolute policy-oracle gap $|C - B|$. Higher values are preferable in (a), while lower values are preferable in (b)-(f). 112

4.9 Policy schedule for Inventory optimisation after 800 training episodes: (a) policy for a single seed and (b) policy consistency with percentage of success for taking consistent actions, across 10 seeds. 113

List of Tables

3.1 Comparisons of different data-driven approaches 54
3.2 Results and Comparison with existing approaches 70
3.3 Robust Vehicle pre-allocation Results 73
3.4 Adaptive Robust Lot-Sizing Results 76

4.1 Hyperparameters used for RiverSwim experiment 99
4.2 Comparison of the ambiguity set centres $\bar{P}$ obtained using Bootstrapped DQN and ϵ-greedy exploration. The columns represent the next state. . 101
4.3 RiverSwim: comparison of robust planning methods for 750 training episodes over 20 independent runs. Entries report the mean ± sample standard deviation. Bold entries identify the best learned method for the corresponding column. As mentioned: $A = \min_{P \in \mathcal{P}} V_P^{\widehat{\pi}}(s_0)$ and $B = V_{P^\star}^{\widehat{\pi}}(s_0)$. 102
4.4 RiverSwim: comparison of robust planning methods for 1500 training episodes over 20 independent runs. Entries report the mean [95% Student-t CI]. Bold entries identify the best learned method for the corresponding column. 103
4.5 RiverSwim: Sensitivity Analysis – Evaluation of different number of K (over 10 independent random seeds for each method) with Bayesian ψ. Each independent run was executed over 750 episodes and for each episode 80 steps. 106
4.6 Hyperparameters used for inventory optimisation experiment 110

4.7 Inventory problem: comparison of robust planning methods for 800 training episodes over 10 independent runs. Entries report the mean ± sample standard deviation. Bold entries identify the best learned method for the corresponding column. As mentioned: $A = \min_{P \in \mathcal{P}} V_P^{\widehat{\pi}}(s_0)$, $B = V_{P^\star}^{\widehat{\pi}}(s_0)$, and $C = V_{P^\star}^{\pi^\star}(s_0)$. 111

4.8 Inventory problem: stress-test evaluation of policies learned after 800 training episodes over 10 independent runs. Entries report the mean ± sample standard deviation. Bold entries identify the largest mean return among the learned methods for each scenario. The oracle policy under $P^\star$ is reported as a benchmark. Stress: $V_{P_{stress}}^{\widehat{\pi}}(s_0)$ 114

List of Notations

Symbol	Meaning
Chapter 3	
u	Uncertain parameter
$\mathcal{U}$	Uncertainty Set in Robust Optimisation
$\mathcal{U}_k$	Uncertainty set associated with component k
x	Initial decision variable in the Robust Optimisation problem
y	Secondary decision variable in the Robust Optimisation problem
c	Objective cost vector
G	Unknown mixing distribution (Random probability in the Dirichlet Process)
G_0	Base distribution of the Dirichlet Process
α	Concentration parameter of the Dirichlet Process
π_k	Mixture weight of component k
μ_k	Mean vector of Gaussian component k
Σ_k	Covariance matrix of Gaussian component k
θ_k	Latent parameters of mixture component k
$N(\cdot)$	Gaussian distribution
$p(\theta)$	Prior distribution
$Gamma(\cdots)$	Gamma distribution
$Beta(\cdots)$	Beta distribution

List of Tables

Symbol	**Meaning**
$W_k(\cdots)$	Wishart distribution
Λ	Precision
μ_0, β_0, Ψ_0, υ_0	Fixed hyperparameters of the base distribution G_0
$\delta(\cdot)$	Delta function
R^m	m-dimensional real vector space
V_k	Stick-breaking random variable for component k
$q(\cdot)$	Surrogate posterior distribution (approximating distribution)
η_k, ν_k	Hyper-parameters for Beta distribution in the Variational inference
N_k	The number of total data points in cluster k
π^*	Threshold to include in the uncertainty set
α_{cov}	Coverage rate (predetermined value representing the desired extent of the uncertainty set)
$\Gamma_{1,k}^{\alpha_{cov}}$	Adaptive scale parameter for cluster k for l_1 norm with α_{cov} coverage rate
$\Gamma_{\infty,k}^{\alpha_{cov}}$	Adaptive scale parameter for cluster k for l_∞ norm with α_{cov} coverage rate
τ_k	Value of the corresponding inverse covariance matrix for each component k
X	Observed dataset
X^*	Bootstrap sample dataset
μ^*	Bootstrapped resampled mean
$\sigma_{\mu_k^*}$	The standard deviation of the bootstrapped resampled means of cluster k
$\rho(\cdot)$	Robust complexity
$ri(U)$	Relative interior of U
t_n	Student's t-distribution with n degrees of freedom

List of Tables

Symbol	**Meaning**
Chapter 4	
$\mathcal{S}$	State space of a Markov Decision Process
$\mathcal{A}$	Action space of a Markov Decision Process
s	State
a	Action
s'	Next state
s_0	Initial state
$r(s, a)$	Reward obtained from taking action a in state s
γ	Discount factor
π	Policy
V_π	Value function of a policy π
P	Transition probability kernel (model)
P^*	True transition kernel (used in simulation evaluation)
$\mathcal{P}$	Ambiguity set over transition kernels
$\mathcal{P}_{s,a}$	Global ambiguity set for the state-action pair (s, a)
$p(\cdot \mid s, a)$	Next-state probability distribution for state-action pair (s, a)
$\bar{p}_{s,a}$	Posterior mean transition distribution for state-action pair (s, a)
$\bar{P}$	Nominal model obtained from $\mathcal{D}$ - Estimated model
$N(s, a, s')$	Number of observed transitions from state s to state s' after taking action a
$\psi_{s,a}$	Ambiguity set radius for state-action pair (s, a)
$V(s)$	Value function at state s
$\hat{V}(s)$	Robust estimated value function at state s
$\hat{\pi}$	Estimated robust policy
P^*	True model
π^*	Estimated robust policy under true dynamics

Symbol	**Meaning**
T	Bellman operator
p^{T}	Transpose
$Q(s, a)$	Action-value function for state-action pair (s, a)
$\mathcal{D}$	Transition sample dataset
$\mathcal{B}$	Replay buffer
d	Metric that measures the distance between two transition probabilities
k	Number of heads
θ_i^k	Q-learning update for each head k
$\bar{\theta}_i$	Fixed target network's parameters
τ	Time steps for stable training
$U(\cdots)$	Uniform distribution
δ	Risk or credibility level used in ambiguity set
i	Step
m_i	Mask bootstrapped for Bernoulli (0 or 1) at step i
ϵ	Probability value between 0 and 1 for ϵ-greedy

List of Abbreviations and Acronyms

Acronym	Meaning
RO	Robust Optimisation
RMDP	Robust Markov Decision Process
MDP	Markov Decision Process
RL	Reinforcement Learning
RRL	Robust Reinforcement Learning
DRL	Deep Reinforcement Learning
DQN	Deep Q-Network
TD	Temporal Difference
DP	Dirichlet Process
DPMM	Dirichlet Process Mixture Model
GMM	Gaussian Mixture Model
VI	Variational Inference
ELBO	Evidence Lower Bound
KL	Kullback–Leibler divergence
LP	Linear Programming
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
ReLU	Rectified Linear Unit
CI	Confidence Interval

Acronym	**Meaning**
SAA	Sample Average Approximation

Glossary of Key Terms

Term	Definition
Uncertainty Set	A set containing plausible realisations of uncertain parameters in RO
Solution Quality	The performance of a computed decision or policy under the stated evaluation criteria
Robustness	The ability of a decision or policy to account for uncertainty
Compact Set	In finite-dimensional spaces, a set is compact if and only if it is closed and bounded. In this thesis, the proposed compact uncertainty and ambiguity sets are additionally designed to fit the represented uncertainty tightly. The comparative phrase "more compact" refers only to this geometric tightness
Ambiguity Set	A set containing plausible probability distributions or transition kernels in RMDP
Tightness	The extent to which an Uncertainty or Ambiguity Set closely represents the uncertainty without including unnecessarily conservative regions
Conservatism	The degree of pessimism or protection induced by an Uncertainty or Ambiguity Set. Broader sets generally produce weakly more conservative worst-case solutions

Term	**Definition**
Nominal Model	The reference estimated model $\bar{P}$ around which an ambiguity set $\mathcal{P}$ is constructed

Acknowledgements

I would like to express my sincere gratitude to everyone who contributed to the completion of this PhD.

First and foremost, I extend my deepest thanks to my supervisors, Prof. Kerem Akartunali and Dr. Bin Liu, for their guidance, support, and insightful feedback throughout the course of this research. Kerem and Bin have always been a tremendous source of support and inspiration for me. I am very grateful to have had their support not only while conducting the studies that led to this thesis, but also in many other aspects of academia. Their encouragement and guidance inspire me tremendously to become the curious researcher I aspire to be today.

I would also like to show my gratitude to the University of Strathclyde for making this PhD thesis possible by providing me with a fully-funded scholarship. I would like to give special thanks for giving me the opportunity to come to the UK and to start a new chapter in my life.

To my colleagues, thank you for the stimulating discussions and encouragement that enriched my PhD journey.

Finally, I am deeply appreciative of my family and close friends for their unwavering patience and belief in me throughout this process. Without their unconditional and sincere love and encouragement, none of this would have been possible.

Publications

Parts of this dissertation have been published in, submitted to for publication, or are in the process of being submitted to for publication in peer-reviewed journals, as listed below:

- Neofytou, A., Liu, B. and Akartunalı, K. (2025), Compact uncertainty sets for robust optimization based on bootstrapped dirichlet process mixture model, *Optimization pp. 1–44*, doi: 10.1080/02331934.2025.2451812
- Neofytou, A., Compact Bayesian Ambiguity Sets for Robust Markov Decision Processes based on Bootstrapped DQN, *(ready for submission)*

I declare that I have led the research and manuscript preparation for the above papers. I contributed approximately 90% of the work, including problem formulation, methodology, implementation, experiments, and drafting. My supervisors contributed the remaining 10%, providing academic supervision, critical feedback, and editorial revisions. The first paper is integrated in *Chapter 3* of this thesis, and the second paper is integrated in *Chapter 4* of this thesis.

Chapter 1

Introduction

1.1 Background and Motivation

A long-standing challenge in decision-making is how to properly account for the uncertainty involved. Specifically, decision-making involves uncertainty in system parameters and model assumptions. In general, decision-making under uncertainty is fundamentally challenged by the lack of precise probabilistic information, computational challenges and the poor performance based on incorrect assumptions (Bertsimas and Thiele, 2006). A common way is to optimise decisions based on a single nominal scenario. However, this can lead to solutions that perform well on paper but fail when reality deviates even slightly from expectations. Given the increasingly complex nature of modern experiments and the large-scale availability of data, various frameworks have been proposed to explicitly address uncertainty. This thesis focuses on the proper uncertainty modelling of two major subcategories of robust decision-making under uncertainty: Robust Optimisation (RO) and Robust Markov Decision Processes (RMDPs). They share the same philosophical foundation, like the min–max formulation and decision-making against worst-case uncertainty, but they can also be applied to different problem structures. We aim to address the previously mentioned issues by seeking solutions that protect against the data and model uncertainty.

Essentially, decision-making under uncertainty can be categorised into two paradigms: static and sequential frameworks, each addressing different classes of problems (Bert-

simas and Thiele, 2006). RO provides a powerful methodology for decision-making under parameter uncertainty, and encompasses static, two-stage, and multi-stage formulations. In a static one-stage formulation, all decisions are made before the uncertain parameters are realised, while in a two-stage formulation, some decisions are made before uncertainty is observed, while recourse decisions are made afterwards. In a multi-stage formulation, decisions are made sequentially over time as uncertainty is gradually revealed (Bertsimas and Georghiou, 2015). For example, in an inventory problem with multiple periods, an initial ordering decision can be made before future demand is known, while later decisions are adjusted after demand is observed in each period. Another example of multi-stage formulation is capacity planning in manufacturing, where we need to decide how to allocate limited machine capacity across multiple products over time. Specifically, future demand for each product may be uncertain, and we need to revise production quantities in later periods as demand information becomes available. This is done by optimising a worst-case objective over an appropriately defined uncertainty set and thus ensuring feasibility across all plausible scenarios (Soyster, 1973). By contrast, in fully sequential decision-making, the RMDP framework extends robustness by computing policies that maximise worst-case cumulative reward under an ambiguity set of state-transition models (Iyengar, 2005; Nilim and El Ghaoui, 2005).

There is a growing body of literature recognising the importance of developing and improving such uncertainty modelling methods. This remains a key research area in both subcategories. Moreover, the heart of those fields is the proper construction of compact and tight data-driven uncertainty and ambiguity sets. This issue has received a considerable critical attention, because it is vital for decision makers, since these approaches influence the final outcome and results. Recent developments in those fields have highlighted the need for data-driven and statistical machine learning methods to improve the performance. Although some research has been carried out on the use of machine learning models to construct data-driven sets, there have been few empirical investigations into statistical and machine learning techniques that attempt to account for the true nature of big data and potential uncertainty involved.

The construction of the uncertainty set in each problem directly shapes the quality

of the resulting decisions. We can easily understand that this makes it a central driver of both performance and robustness. Consequently, this critical influence has motivated a growing body of work aimed at constructing novel and data-driven approaches that surpass the limitations of traditional state-of-the-art methods. Building on this motivation, this thesis develops and evaluates innovative machine learning–based frameworks for constructing compact uncertainty and ambiguity sets, with the goal of enhancing decision-making under uncertainty in both RO and RMDP.

Therefore, while robustness is vital in several practical domains that face uncertainty, robust decision-making guarantees performance even in worst-case scenarios. Consequently, a key aspect of robust decision-making is distinguishing between different sources of uncertainty, including data uncertainty, parameter uncertainty, and model uncertainty. Data uncertainty implies noise or errors in the available data used to estimate model parameters. Parameter uncertainty arises when the structure of the decision model is assumed to be fixed, but the values of its parameters are uncertain. In contrast, model uncertainty refers to uncertainty about the specification or structure of the model itself. Thus, parameter uncertainty can be viewed as uncertainty within a given model specification, while model uncertainty implies whether the specification itself is appropriate. Both types of uncertainty can degrade solution quality, while data uncertainty can further affect the estimation of model parameters. In the next section, we will explain the holistic idea of a Decision-Making process, how uncertainty affects the outcome and why robust decision-making frameworks are vital to generate robust decisions.

1.2 Decision-making process

Here, we will provide to the decision-making process a brief introduction to the elements that generate a "good" outcome. In Figure 1.1 we present the fundamental process, which primarily involves Data. Next, we have an actual Model, meaning a framework which we can think about or analyse the system. Then, it involves the Optimisation part, where the Data and the Model are brought together in a coherent way in order to create the analysis. Ultimately we lead to Decisions, where the value of this whole

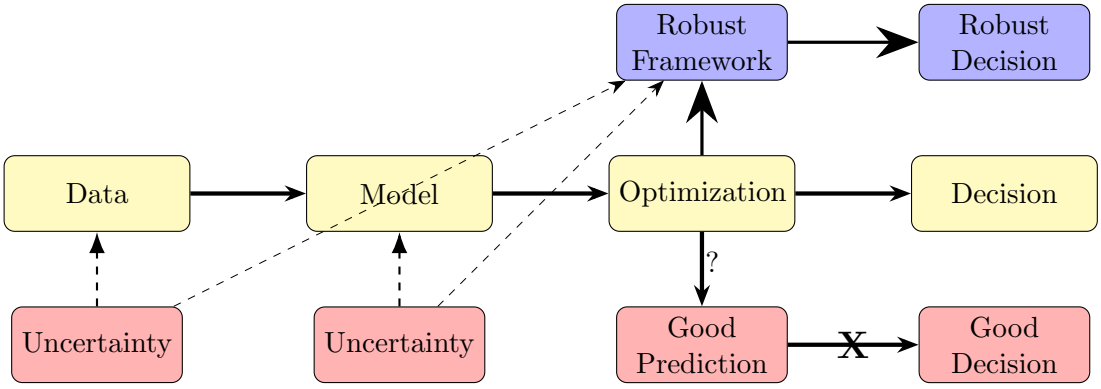


Figure 1.1: Decision-Making under uncertainty framework

process is presented.

However, in real life scenarios, there is usually some kind of uncertainty involved in the Data and Model steps. If we do not account these and we proceed to the standard Optimisation part, we make predictions that we initially believe are "good" enough, however, they often do not lead to "good" Decisions. This would lead to catastrophic results and outcomes in various critical domains. To address this gap, a robust framework should be introduced, which explicitly accounts for the previous uncertainties and allows the optimisation process to produce decisions that are mainly robust but also high-quality.

We emphasised and highlighted the necessity of such robust frameworks in bridging the predictive-prescriptive gap in decision-making under uncertainty. Depending on the context, the decision making framework should be either static or sequential. In the next subsections, we will focus on the vital role of the uncertainty modelling in the two key areas: RO and RMDP. Specifically, we will explain why we focus on uncertainty sets for both decision-making problems and their importance in the resulting solution quality and robustness.

1.3 Uncertainty: Types and Modelling

As mentioned above, in decision-making problems we have different types of uncertainty involved. Data uncertainty can also be seen as statistical uncertainty and refers to inherent randomness or some type of noise in observed data. Even if the form of

the model is correct, limited or noisy data cause uncertainty in estimating the vital parameters. On the other hand, model uncertainty can also be seen as epistemic uncertainty and refers to uncertainty about the actual model itself. For instance, we may have unknown or misspecified dynamics, which mainly occurs when the true generating process is only partially known. This type of uncertainty is critical, as optimal policies can be highly sensitive to slight changes or errors in model parameters. Once again, this highlights the need for robust frameworks that account for these types of uncertainties.

The decision-making problems we are examining require decisions to be made before the above uncertain quantities, that affect the outcome, are fully observed. Let $x \in \mathcal{X}$ denote a decision vector and let $u \in \mathbb{R}^m$ denote an uncertain parameter, such as demand, cost, or return. A nominal optimisation model typically replaces u by a point estimate $\hat{u}$ and solves $\min_{x \in \mathcal{X}} f(x, \hat{u})$. However, if the realised value of u differs significantly from $\hat{u}$, the resulting decision will become infeasible, unsafe, or highly suboptimal, depending on the domain. This is the basic and general related uncertainty problem considered in this thesis.

In the next subsections, we will emphasize how the uncertainty modelling, and specifically the proper uncertainty set construction, influence the outcome in both key areas, by also presenting their formulations separately. We essentially acknowledge that uncertainty in data leads to an ambiguous model, which requires cautious decisions. Motivated by these considerations, we explicitly focus on RO and RMDP frameworks by planning for worst-case deviations.

1.3.1 Uncertainty Sets in Robust Optimisation

RO addresses uncertainty by ensuring solutions that remain feasible for all parameter realisations within a predefined uncertainty set. Consequently, the choice of this uncertainty set determines the balance between robustness and solution optimality. However, traditional uncertainty sets are usually fixed in shape and size. This results in a misrepresentation of the true nature of uncertainty. Overly large uncertainty sets lead in unnecessarily conservative solutions, because they sacrifice optimality, while overly small sets risk infeasibility when an unexpected parameter realisations occur.

This limitation has driven research towards data-driven uncertainty sets. Novel methodologies are developed that are informed directly by observed data and allow uncertainty sets to remain compact and tight, but still ensure robust decisions. This evolution highlights the need for exploring further advanced and data-informed approaches, but also the importance of uncertainty modelling to balance conservatism and performance. Generally, one-stage RO is a framework suitable for static decision-making contexts, where decisions x are made once and must remain robust under uncertainty without further adjustments. However, there is also a two-stage RO setting, which is not entirely static but has very limited decisions.

Parameter uncertainty implies the uncertainty in the values of the parameters that define the optimisation problem. In this context of RO, this uncertainty directly affects the feasibility of solutions, since the true parameter values can deviate from their nominal estimates. Instead of assuming that these parameters are exactly known, several authors attempt to construct novel uncertainty modelling techniques, which acknowledge noisy datasets or some estimation error.

The RO formulation protects the decision against this set of plausible uncertainty realisations $\mathcal{U}$. The generic robust counterpart can be written as:

$$\begin{aligned} &\min_{x \in \mathcal{X}} \max_{u \in \mathcal{U}} f(x, u), \\ &g_i(x, u) \leq 0, \qquad \forall u \in \mathcal{U}, \quad i = 1, \ldots, p. \end{aligned} \tag{1.1}$$

This thesis addresses parameter uncertainty by aiming to construct such data-driven uncertainty sets directly from empirical observations of the uncertain parameters. The target is to construct uncertainty sets that are less conservative than traditional methods. Essentially, we plan for all parameter realisations within the constructed set and also ensure that the resulting decisions remain feasible.

1.3.2 Ambiguity Sets in Robust Markov Decision Processes

RMDPs extend robust optimisation to fully sequential decision making. We are essentially addressing uncertainty in state-transition probabilities. Once again, a central aspect to RMDPs is the definition of an uncertainty set, commonly called ambiguity

set. This means capturing plausible variations in the underlying transition dynamics.

However, existing ambiguity set construction methods produce overly conservative solutions by relying on fixed datasets with limited coverage of critical state–action pairs. Furthermore, traditional RMDP ambiguity sets fix a nominal transition model P and essentially define deviations around it. Today, data-driven methods address this by dynamically optimising the most important elements of an ambiguity set: centre and size. The target is to find suitable methodologies to construct uncertainty representations that ensure robust policies but still provide guarantees without being excessively pessimistic.

Distributional uncertainty implies the uncertainty about the underlying probability distributions that generate uncertain quantities and observed data. In contrast to parameter uncertainty, distributional uncertainty refers to the probability distribution itself. In the context of RMDPs, it specifically refers to uncertainty about the transition probabilities between states. Rather than assuming these probabilities are precisely known, we aim to find a distributionally robust approach that acknowledges that the true distributions might differ from nominal distributions. As mentioned before, this is due to limited or noisy data, and model misspecifications.

Specifically, the estimate $\bar{P}$ may be inaccurate. RMDP therefore considers the ambiguity set $\mathcal{P}$ of plausible transition kernels and seeks a policy that performs well under the worst-case transition model:

$$\max_{\pi} \min_{P \in \mathcal{P}} \mathbb{E}_{P,\pi} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right] \tag{1.2}$$

Our thesis addresses this distributional uncertainty by aiming to construct such ambiguity sets directly from observed data-driven transitions. Our model is fundamentally a distributionally robust approach, because it explicitly recognises that the true distribution may differ from the estimated one due to limited data and we thus incorporate this uncertainty directly into the ambiguity set construction process.

1.4 Main Contributions

The scope of this thesis lies at the intersection of statistical methods, data-driven approaches and machine learning techniques within RO and RMDP. Both fields can be characterised as complementary frameworks within the broader domain of robust decision-making under uncertainty. However, several critical challenges and limitations persist. Particularly, decision-making under uncertainty faces significant challenges across various practical domains, including supply chain management (Akartunalı and Dauzère-Pérès, 2022; Attila et al., 2021). Traditional robust decision-making methods rely heavily on simplified and fixed uncertainty assumptions. These are theoretically rigorous; however, these simplistic approaches frequently generate overly conservative solutions, which limit practical effectiveness in real-world settings.

Recent advances in data-driven machine learning offer a prospect to properly address these limitations. One direction will be to utilise the observed data, where machine learning approaches allows the construction of uncertainty and ambiguity sets that are both compact and accurately reflect the actual uncertainties. Nevertheless, existing efforts by several authors in the literature still lack a data-adaptive framework for the proper uncertainty set construction. In general, the uncertainty set construction focused using innovative and advanced machine learning-based techniques remains unexplored.

Motivated by these opportunities and gaps, this thesis seeks to advance robust decision-making methodologies through innovative integrations of uncertainty modelling techniques. Our work is already theoretically supported by the existing literature and provided references, while we explicitly focus on the practical and methodological contribution which are active and key research directions. Specifically, our main target is to improve the trade-off between robustness and optimality, which is vital because it leads to less conservative decisions. This has a big impact in various fields, which seek safety and guarantees. Furthermore, this should improve the performance of robust decision-making across RO and RMDP settings. Even though we will present the research questions for each chapter individually at the later stage, below we provide

the research objectives and aims of this thesis.

Research Objectives

RO.1. Develop a data-driven method to construct compact uncertainty sets for RO that improve the robustness–optimality trade-off

RO.2. Use nonparametric Bayesian mixture modelling to represent complex uncertainty structures and integrate them in the uncertainty set

RO.3. Improve the outcome of the estimated parameters of the uncertainty set using bootstrap resampling

RO.4. Develop a data-driven RMDP ambiguity set using Bayesian credible regions that remains tractable in robust planning

RO.5. Integrate Bootstrapped DQN variant exploration to adjust the centre and the radius of the compact ambiguity set

RO.6. Empirically evaluate the impact of the proposed data-driven uncertainty and ambiguity sets against relevant baselines, mainly in terms of objective value (RO) and worst-case return (RMDP)

Kindly note that RO.1, RO.2, RO.3 and RO.6 are addressed in Chapter 3, while RO.4, RO.5 and RO.6 are addressed in Chapter 4. Our contributions in this thesis can be summarised as follows, listed by chapter:

Chapter 3: Compact Data-Driven Uncertainty Sets for Robust Optimisation

- In this chapter, we propose a modified combination of an unsupervised learning technique, variational inference, stick-breaking and outlier detection to estimate the parameters of the Gaussian Mixture Model (GMM) from observed data. Variational inference enables efficient approximation of the posterior distribution, while stick-breaking simplifies the process of allocating clusters within the GMM. Additionally, the inner model's outlier detection identifies the influence of anomalous data points, ensuring robust parameter estimation.

- We employ bootstrap resampling to further reinforce the robustness of the exported parameters. It essentially provides a way to estimate the uncertainty of the clustering results. Consequently, we obtain more robust estimates and improve the overall performance of our approach.

- For the data-driven uncertainty set construction, to ensure the desired level of conservativeness in the solution, we adapted a technique to compute the minimum adaptive scale parameter. With that we managed to capture the variability in uncertainties and integrated it in our uncertainty set. In addition, we showed that the final constructed uncertainty set is given by a finite union of convex subsets and their intersected norms.

- We tested our approach under various experimental settings and data structures to evaluate its comparative performance against the selected benchmark methods in the considered (two-stage) RO problems. Additionally, we evaluated the performance of the proposed data-driven uncertainty set in comparison with traditional state-of-the-art and data-driven methods in terms of objective value, robustness, computational time and different coverage rates. The findings of this study clearly show that the constructed uncertainty set provides a more decent objective compared to alternative sets, while we still retain its robustness.

Chapter 4: Compact Bayesian Ambiguity Sets for Robust Markov Decision Processes

- In this chapter, we present a data-driven methodology for constructing ambiguity sets by using simplified online Bootstrapped DQN variant exploration to define the optimal location (centre) for RMDP. Unlike traditional approaches that mainly rely on empirical averages, and in contrast to offline RMDP settings where the dataset is fixed, our approach employs deep exploration trajectories to form more robust and compact ambiguity sets. We are essentially learning the ambiguity set from data we actively collect. This allows us to improve the quality of the exported robust policies under uncertainty.

- To overcome the conservativeness issue, which is common in classical ambiguity set constructions, we adopt a Bayesian model using Dirichlet distributions to compute the ambiguity set radius (size). Specifically, we employ posterior sampling based on the previously robustly collected transition probabilities. This innovative integration with the conducted deep exploration leads to more adaptive sizes for ambiguity sets, but also significantly reduces overly conservative outcomes for specific state-action pairs (s,a). Consequently, this generates policies that could better balance risk and performance.

- We rigorously validate our proposed method through two detailed numerical experiments. Results from these studies demonstrate the improved performance of our approach in comparison to traditional exploration strategies and conservative ambiguity sets. We present that the resulting robust policies effectively account uncertainty, while also achieving closer-to-optimal worst-case returns and consistently outperforming traditional state-of-the-art methods.

1.5 Thesis Structure Overview

This thesis consists of 5 main chapters. Following the introduction provided under this chapter, a literature review on the main research areas is given in the Chapter 2. In Chapter 3, RO and data-driven uncertainty set construction are formally stated. In addition, we present our integrated methodology that generate uncertainty sets with improved performance than classical state-of-art methods. Chapter 4 is dedicated to the ambiguity set construction in RMDP, along with our innovative integration that provides significant improvement in terms of overall performance compared to other traditional methods. Finally, the vital conclusions, discussions and interesting future research directions are presented in Chapter 5.

Chapter 2

Literature Review

In this chapter, we provide a literature review on the two key areas we are investigating: RO and RMDP. Specifically, we will focus on their applications and the motivation for innovative state-of-art approaches. We commence our study of robust decision making under uncertainty with the classic RO setting (static and two-stage), while also describe traditional uncertainty sets. We then present a review on the integration of machine learning, what was done by other researchers and discuss recent methodologies that aim to tackle previously mentioned challenges. Next, we extend our robust decision-making under uncertainty setting to sequential problems and provide an overview of studies with related works in RMDPs. Furthermore, we discuss and outline some related works in exploration and its prospect in the ambiguity set construction. Finally, we emphasize the critical gap and vital issues that persist in both research fields.

2.1 Robust Optimisation

2.1.1 Classical robust optimisation and uncertainty sets

One of the most effective static decision-making frameworks is RO, which has emerged specifically to handle uncertainty. This is accomplished by ensuring solutions remain feasible across all plausible scenarios within predefined uncertainty sets. Early contributions mainly aimed at ensuring this feasibility under worst-case scenarios. Particularly, a highly influenced work was introduced by Soyster (1973). In his seminal work and

formulation, Soyster proposed using a "box" uncertainty set, a simple interval, where uncertain parameters independently varied within some fixed ranges. This research provides strong robustness guarantees; however, these uncertainty sets tend to produce highly conservative solutions. The reason is that they implicitly assume that all parameters simultaneously take their most extreme values (Soyster, 1973).

Towards the end of the 20th century and clearly motivated by the previous limitations, Ben-Tal and Nemirovski (1998) introduced ellipsoidal uncertainty sets to RO in 1998. These ellipsoidal sets allowed uncertainty to be modelled with a more realistic correlation structure among the parameters. At that time, this innovation significantly reduced conservatism (Ben-Tal and Nemirovski, 1998). However, the choice of appropriate ellipsoidal dimensions remained extremely challenging, mainly due to inaccurate assumptions regarding correlations. This introduced non-linear function to the model and thus computational challenges by do not capturing the correlated information due to the symmetric shape. Ben-Tal and Nemirovski (1999) subsequently developed and formulated robust counterparts for uncertain linear programs, which require hard constraints to hold for all parameter realisations within a prescribed uncertainty set.

At the start of the 21st century, Bertsimas and Sim (2004) proposed "budgeted uncertainty sets" in 2004. The target was to further address the trade-off between robustness and solution conservatism that was highlighted by the previous study. Specifically, the authors presented a polyhedral approach that allowed a controlled conservatism, primarily through a parameter known as the uncertainty budget. Unlike the earlier box model, which considered all uncertain parameters at extremes simultaneously, this method permitted only a subset of parameters to reach worst-case deviations simultaneously. Consequently, this introduced the flexibility and improved practical performance (Bertsimas and Sim, 2004) that the literature was lacking, by controlling the level of robustness and thus balancing protection against uncertainty. However, this diamond shape was more suitable for independent uncertainty. In parallel, Ben-Tal et al. (2004) extended RO to uncertain linear programs in which some decision variables are fixed before uncertainty is revealed, while others can adjust afterward.

For worst-case and predefined uncertainty sets, several attempts have been made

with norm bounds, l_1 or l_∞ norms. Prior research has typically been limited to offering empirical validation of model robustness in constrained situations (Shang et al., 2017). As mentioned, other methodologies involved box set, ellipsoidal set, bounded by norms or polyhedral uncertainty sets (Bertsimas et al., 2021; Faccini et al., 2022). However, these methods on their own of l_1 or l_∞ norms have several limitations, such as unable to capture the specific shape of uncertainty, as it only considers the magnitude of the individual parameters for l_1 norm, as well as l_∞ norm may not capture correlations beyond the largest individual parameter (Bertsimas et al., 2021).

Collectively, these classical contributions laid the theoretical and methodological foundations of RO. However, their fixed and simplistic uncertainty representations failed miserably to accurately capture complex real-world uncertainty structures, which are especially obvious nowadays. Over the past years there has been a significant increase in the interest to improve the uncertainty set construction using more advanced techniques and tools.

One explanation for this shortcoming is that the previous classical RO methods traditionally relied on predetermined assumptions without strictly requiring data. Nevertheless, recent works strongly emphasize the critical role of data and their vital role in today's dynamic. To significantly enhance the solution accuracy and reducing unnecessary conservatism, which is one of the main objectives in the field, data-driven RO methods utilise historical data to construct more realistic uncertainty sets. Consequently, the rise of unsupervised learning and clustering techniques has been increasingly recognised in this context and several authors try to adapt their application in the specific area of uncertainty set construction. The target aim is to construct uncertainty sets that more precisely reflect underlying patterns and correlations observed in data. This thesis is motivated by the previous limitations for an ongoing evolution toward more data-driven and machine learning approaches in RO's uncertainty modelling, as discussed in subsequent subsection.

2.1.2 Machine Learning in the data-driven uncertainty set construction for robust optimisation

In the context of RO and data-driven uncertainty set construction, unsupervised learning techniques play a vital role (Goerigk et al., 2023). The target is to discover patterns and relationships within unlabelled data, without any guidance (Jordan and Mitchell, 2015), and to group similar data points, in order to construct relevant uncertainty sets for each cluster (Blei and Jordan, 2006). Additionally, unsupervised anomaly detection techniques supports this process by identifying outliers, which is vital for handling extreme cases in uncertainty modelling (Zhu et al., 2016).

Instead of enhancing the algorithmic performance (Bertsimas and Georghiou, 2015; Dumouchelle et al., 2023; Goerigk and Kurtz, 2022; Julien et al., 2022), diverse machine learning models have been applied to generate compact uncertainty sets from historical data that lead to the learning of high quality representations, which is the aim of our research. In Goerigk and Kurtz (2023) an unsupervised deep learning method is utilised to construct non-convex uncertainty sets and the trained neural networks was integrated into the RO model. Their uncertainty set was defined as a sphere of radius R and they trained their neural network employing one-class classification, which minimises the empirical centred total variation of the projected data points (Ruff et al., 2018). After completing the training of the network, adjustments are made to the radius R of the uncertainty set with the aim of achieving a desired coverage of the dataset. Some other ideas use principal component analysis and kernel smoothing in Ning and You (2018) for similar goal. In Shang and You (2019) the novel idea was to apply support vector clustering and in Bertsimas et al. (2018) statistical hypothesis testing to construct the data-driven uncertainty sets. Attempting to bridging the gap between machine learning and RO, Shang et al. (2017) proposed a data-driven approach for the construction of an uncertainty set using kernel-based support vector clustering. However, kernel methods can face significant challenges when applied to high-dimensional data (Shang et al., 2017), where the computational cost grows significantly and leads to overfitting. To tackle the decision-making under uncertainty and find a balance between robustness and performance, the authors in San Juan and Sy (2024) used machine learning

predictive models to guide the optimisation process. In Arabsheybani et al. (2024), the authors utilised clustering algorithms to analyse historical data, identify patterns and thus create clusters that represent different scenarios of uncertainty and, consequently, generate uncertainty sets.

Exploring all the possible unsupervised machine learning algorithms, a relevant work in Burgard et al. (2022) applied k-means to hedge the clustering result against unstructured errors in the observed data. k-means is the most well known unsupervised method in the machine learning world and we aim to compare our results with that. However, it has several limitations, such as when the data points in clusters overlaps other clusters, making this approach not performing well. Furthermore, the number k, the number of clusters, should be determined beforehand which means that different initialisation can lead to different final cluster assignments and centroids. It heavily relies on the original data representations and thus suffers from the curse of dimensionality (Bertsimas and Kim, 2024; Jordan and Mitchell, 2015). Also, k-means may not perform well when the clusters are non-spherical since it assumes that the data points are distributed in a spherical shape (Burgard et al., 2022).

In Wang et al. (2022), the authors addressed uncertainty in optimisation problems by focusing on the expected value of the uncertain parameters. They applied clustering and ensured that solutions are optimised against average and weighted scenarios rather than worst-case extremes. The result was less conservative solutions by utilised k-means for their unsupervised learning approach. The primary drawback of this mean RO approach is the accurate estimation of mean values. Furthermore, in their experimental settings they compare different number of clusters for the k-means and they determine the optimal number of clusters with the elbow method to compare the objective. However, they need to define the optimal number of clusters beforehand for every scenario.

In Dai et al. (2020), the authors attempted to create two data-driven polyhedral uncertainty sets that directly utilise l_1 and l_∞ norms respectively for their own purpose. Their data-driven uncertainty set was a union of intersected polyhedral uncertainty sets, where each specific norm uncertainty set covers the data belonging to each cal-

culated component. As a result, it forms the final uncertainty set with the union of all constructed l_1 and l_∞ norms. A similar but more innovative attempt was made by Ning and You (2017) with the intersection of the two l_1 and l_∞ norms. The suggested polyhedral uncertainty set comprises again a combination of basic uncertainty sets, extending beyond the constraints of a 'parallel-piped' set. Subsequently, the resultant robust model produces solutions of exceptional quality. Instead of the l_1 uncertainty which relies solely on the l_1 norm, this new data-driven uncertainty set is defined using both the l_1 norm and l_∞ norm. When the intersection of the two norm balls are constructed, the final data-driven uncertainty sets differ and lead to distinct optimisation solutions due to the intersection. Both works adapt the data-driven uncertainty set for their data using Dirichlet Process Gaussian Mixture Model as the unsupervised learning algorithm in order to calculate the important parameters and Variational Inference to approximate the posterior distribution of latent variables. This is because the exact inference for the mixture parameters was intractable, thus their uncertainty set construction relied on those approximated parameters and predefined probability guarantees.

In Li et al. (2022) the authors challenge the above model's adaptability and clustering. They tried to improve the performance of the Dirichlet Process Mixture Model (DPMM) with the introduction of deep neural networks in a Deep DPMM. This is a deep unsupervised method that simultaneously performs clustering and learn more expressive features. However, they employed expectation-maximisation, instead of Variational Inference, and used Gibbs sampling to sample from the posterior. In Kimura et al. (2013) apart from RO, the authors introduce recursive expectation-maximisation algorithm ideal for parameter estimation in DPMMs. The algorithm offers significantly improved computational efficiency compared to Variational Bayes and Inference methods but with its own limitations. Deep unsupervised learning was also employed in Goerigk et al. (2023) to learn and extract hidden structures and anomalies from data with non-convex uncertainty sets for more robust solutions.

To conclude, most works focused on innovative approaches that integrate unsupervised learning with RO to enhance decision-making under uncertainty. Specifically,

the above clustering methods are widely used to identify hidden structure in uncertain data, but they involve several limitations. In this thesis, we will use DPMM, which was chosen not only because it avoids fixing the number of clusters in advance, but because it gives a probabilistic Bayesian mixture model with a clear cluster assignment and posterior uncertainty. With that we obtain efficient and specific cluster probabilistic descriptions, including means, covariance, weights, all of which are vital for constructing data-driven uncertainty sets in RO. DPMMs provide these quantities naturally, whereas several alternative clustering methods do not provide a direct Bayesian posterior representation. Furthermore, Gaussian components are ideal to model clusters with different sizes and shapes, while the exact inference for the mixture parameters is intractable, thus its uncertainty set construction heavily relies on those approximated parameters. In addition, the concept of just accounting exactly all the uncertain data points and scenarios does not safeguard again data perturbations, uncertain data collection, noise or parameters deviation. Here, perturbations refer specifically to changes or deviations in the observed uncertain data, including noise and newly observed data patterns. Consequently, this may alter the above estimated means, covariances, mixture weights and thus resulting uncertainty set geometry. Therefore, we employ bootstrap resampling to reduce sensitivity to such sampling variation. We will take it a step further and attempt to further improve the robustness of solutions, challenging the concept of balancing optimality and robustness. Our novelty and idea are to use this resampling method in the uncertainty set construction, that to the best of our knowledge has not been employed before. This provides more reliable estimates of parameters by generating multiple samples from the data (Bertsimas and Van Parys, 2022). While the predictive analytics phase of unsupervised machine learning helps understand the data and make predictions about the underlying structure, the prescriptive analytics phase of RO and uncertainty set construction will utilise these insights not only to prescribe optimal decisions, but also robust decisions in the face of uncertainty. To this end, our work seeks to address current challenges and limitations in uncertainty modelling by using clustering to construct reliable nonparametric data-driven uncertainty sets for RO problems.

While RO provides a leading framework and a powerful tool for handling parameter uncertainty, it is typically applied to static or finite-stage decision problems. Several real-world settings, however, involve fully sequential decision processes. To address such problems, it is necessary to adopt a framework that models such sequential interaction with an uncertain environment. Therefore, the next section introduces the Reinforcement Learning (RL) setting, specifically the RMDP, and the gaps that continue to persist.

2.2 Robust Markov Decision Processes and Reinforcement Learning

2.2.1 Related studies in Robust Markov Decision Processes

The key challenge in the area of RL is that traditional methods assume a fixed environment model (Sutton, 2018), which implies inefficient performance. This also leads to poor generalisation when inconsistencies arise between the model and real-world conditions. In addition, we want to ensure that the learned policies remain robust to inherent uncertainties and potential parameter variations. There are several works for tackling parameter uncertainty in Markov Decision Processes (MDPs), with contributions from Satia and Lave (1973) and later works by Xu and Mannor (2010) and Nilim and El Ghaoui (2005).

Unlike standard MDPs (Baird, 2020; Lim et al., 2013; Puterman, 2014; Sutton, 2018), RMDPs account for uncertainty in transition probabilities. These parameters vary within an uncertainty set, what we call an ambiguity set in Robust Reinforcement Learning (RRL), which has all the values for transition probabilities (Petrik and Russel, 2019; Petrik and Subramanian, 2016; Wiesemann et al., 2013). The concept of ambiguity sets in RRL is closely related to uncertainty sets in RO (Ben-Tal et al., 2009; Bertsimas et al., 2018; Brown et al., 2020). In RO, the goal is to find solutions that perform well across all distributions within the uncertainty set (Russel et al., 2019). As a result, the effectiveness of an optimal RMDP policy directly reflects on the structure of that ambiguity set. This implies that it should be small enough to ensure that the

solution is a lower bound, but also large enough to allow robust exploration (Petrik and Russel, 2019). Those data-driven ambiguity sets are usually constructed as confidence regions informed by concentration inequalities. Nevertheless, these concentration inequalities result in overly conservative solutions (Auer et al., 2008; Ghavamzadeh et al., 2016; Thomas et al., 2015; Weissman et al., 2003) and that is why the vital challenge is to propose ambiguity sets that do not rely on confidence regions.

The literature in the area of Distributionally RMDP has grown extremely fast. To address the previous issue and gap, we have to assume that the inherent uncertainties can take on any value from the pre-defined ambiguity set, with initial contributions from Iyengar (2005); Wiesemann et al. (2013); Xu and Mannor (2009) and later evolved works by Lu et al. (2024); Wang et al. (2023). In general, we can use domain knowledge or priors to shrink ambiguity sets appropriately (Petrik and Subramanian, 2016), but this is challenging. As a result, in Petrik and Russel (2019), the authors examined the construction of ambiguity sets for RMDPs using Bayesian methods. The authors challenged the issue of traditional confidence regions that lead to conservative policies and presented tight Bayesian ambiguity sets that better capture uncertainty. According to Petrik, this introduction of Bayesian methods to update the ambiguity sets led in an improvement in computing robust policies, assuming a known true environment. Essentially, instead of a fixed confidence level for all state-action pairs, Petrik's method properly adapts the ambiguity set based on the posterior distribution such that it produces an appropriate size for the ambiguity set depending on the data or the prior. In general, Bayesian approach is a promising direction, but standard RMDP works assume given models and Petrik relied on a fixed dataset in an offline batch setting. As a result, this method highly depends from the quality of the transition data.

The call to explore different approaches to construct tight ambiguity sets led researchers to further explore Bayesian methods, because these provide a natural way for quantifying uncertainty. Unlike traditional methods, Bayesian models use prior knowledge and adaptively update ambiguity sets based on observed data. Derman et al. (2020) proposed a Bayesian approach called Uncertainty Robust Bellman Equation which tackles the issue of overly conservative solutions in RMDPs. In addition,

diverse methods have been addressed to overcome the conservative nature of resulting policies. The authors in Russel et al. (2019) performed norm-bounded weighted ambiguity sets. This accounted the ambiguity set's shape and its weight based on the structure of the value function. Russel showed that these weighted policies were superior from traditional methods in various domains. Furthermore, the authors in Goyal and Grand-Clement (2022) have questioned why to use sa-rectangular assumption, thus they introduced a d-rectangular assumption with a focus on the computational challenges.

For reducing uncertainty in RMDPs, Gadot et al. (2023) used a method to balance performance and robustness. Also, Panaganti et al. (2022) presented a model called Robust Fitted Q-Iteration, which was suitable to learn robust policies using offline data. Policy approximation is widely used in the field, thus the authors in Panaganti and Kalathil (2022) designed control policies that are robust against uncertainties in the simulator model compared to real-world settings. They proposed a model-based RL algorithm called Robust Empirical Value Iteration. Their aim was to learn an approximately optimal robust policy even when the nominal model is unknown, considering three classical types of uncertainty sets: total variation, chi-square, and Kullback-Leibler divergence. On the other hand, in Derman et al. (2023), the authors investigated how regularisation methods can be used to achieve better robustness in MDPs with uncertain transitions. In Neofytou et al. (2025) the authors constructed compact uncertainty sets, but for a RO setting, based on a Bayesian model and bootstrapped resampling.

2.2.2 Related studies in Exploration

Efficient exploration continues to be a key challenge in RL. In general, the optimal exploration strategy for priors over MDPs can be determined through dynamic programming (Osband et al., 2016). Still, obtaining the exact solution is not feasible and thus the solution is to emphasize in generalisation and approximate methods. This highlights the need for efficient robust exploration methods (van Hulst et al., 2025), so diverse online exploration strategies have been proposed with the aim to lead to useful

learning.

One of the earliest heuristics studies involving Bayesian sampling approaches was Thompson Sampling in Thompson (1933), originally developed by Thompson in the context of multi-armed bandit problems. Essentially, it involves drawing a single sample from the posterior distribution at each time step and selecting the action that seems the best based on that sample. In RL settings, the agent should draw a sample from the posterior of the value function, but resampling at every time step is quite inefficient.

To balance exploration and exploitation effectively, the authors in Osband et al. (2019) introduced the concept of Randomized Value Functions to guide deep exploration. Traditionally, the exploration in a complex environment is performed with a simple ϵ-greedy method, but this is quite insufficient. Still, modifying DQN to improve exploration has became extremely popular. As a result, there was a tremendously evolved work by Osband et al. (2016), where the authors applied bootstrapping to generate a variety of Q-value estimates in order to promote deep exploration. The challenge is that the dropout approach was not suitable to generate effective uncertainty estimates for some tasks. Consequently, the authors used bootstrapped Q-value heads to address the exploration-exploitation trade-off. Without the bootstrapped masking where the mask is all 1's for every data point, the method becomes an ensemble DQN (Osband et al., 2016; Owen and Eckles, 2012). Nevertheless, Osband observed that some of the Bootstrapped DQN's benefit came from the initial random diversity of the ensemble heads. An extension of this work in Meng et al. (2022) attempted to improve the diversity of Bootstrapped DQN algorithm by replacing priors with noise. To the best of our knowledge, the studies from Osband et al. (2016, 2019) were some of the most promising attempts in the research direction of smart exploration. However, the prospect of such an innovative online exploration technique has limited use in a model-based setting and particular in the construction of tight ambiguity sets, which is quite underexplored.

Consequently, the construction of less conservative policies has been recognised as a vital challenge in RL. Nevertheless, constructing such high-quality policies using RMDP and robust ambiguity sets is challenging. Several authors explored diverse approaches

and introduced methods to balance robustness and performance, focusing on both the size and location of those sets. Bootstrapped DQN was originally introduced to assist in deep exploration in RL settings (Osband et al., 2016), but it has not been previously integrated as a tool to further support the design of the robust policy and the construction of tighter compact ambiguity sets.

To the best of our knowledge, the introduction of such innovative method to improve the model-based RL setting of ambiguity set construction in RMDP, along with with Bayesian modelling, is not widely used and has limited attention. Our work aims to bridge this gap. RMDP studies have largely treated the centre and radius of ambiguity sets in isolation, while also relying on given datasets, simple empirical means or confidence regions (Petrik and Russel, 2019). To date, no work has combined deep bootstrapped exploration, to position the set, with Bayesian posterior, to size the set, within this setting.

2.3 Research Gaps and Opportunities

As shown, there is a significant progress in decision making under uncertainty with the use of machine learning models in both RO and RMDPs. However, several critical research gaps remain which involve proper uncertainty set construction methods with diverse data-driven techniques that account the uncertainty and thus provide both robust and high-quality outcomes.

Classical RO and RMDP approaches rely on simplistic and predefined uncertainty representations. These traditional methods generate overly conservative solutions and policies, even though they guarantee computational tractability and worst-case protection. This results in reduced practical effectiveness in real world scenarios and experiments. Recent literature acknowledges the importance of data-driven methodologies, but existing works on data-driven methods remain limited. Another significant gap is in the proper integration of even more advanced and underexplored machine learning techniques, such as Bayesian nonparametric models or deep exploration before the uncertainty and ambiguity set construction. Current approaches barely combine such statistically robust learning models into the formulations. This lacks the proper adap-

tion to the data, which includes the uncertainty, and potentially limits the practical effectiveness of the robust decision-making methodologies.

This thesis aims to develop and evaluate machine learning methods for both compact uncertainty and ambiguity set construction. We will strongly emphasize on reaching both compactness and accuracy. By doing so, we seek to significantly enhance the robustness and performance trade-off in both RO and RMDP contexts. Finally, we aim to critically evaluate and advance existing works, while also contributing original and practically impactful methodologies to the robust decision-making literature.

Chapter 3

Compact Data-Driven Uncertainty Sets for Robust Optimisation

We begin our in-depth analysis on decision-making under uncertainty with a static context, focusing on classical RO. We then expand on two-stage RO, which is not as richly sequential and has limited decisions. Our main target is to understand how statistical and machine learning models can improve the solutions by influencing the main component of RO: the data-driven uncertainty set construction.

3.1 Introduction

In today's dynamic and uncertain environments, the effective utilisation of machine learning techniques has become increasingly important (Jordan and Mitchell, 2015; Qin, 2014). These methods have been applied in numerous applications across various domains, including supply chain management among others (Ben-Tal and Nemirovski, 2000; Garcia and You, 2015). The integration of machine learning algorithms allows organisations to utilise enormous amounts of data to gain valuable insights and make informed decisions (Jordan and Mitchell, 2015). However, traditional optimisation methods often face significant challenges in handling the inherent uncertainties present

in real-world scenarios (Sahinidis, 2004). Recently, to overcome these challenges and harness the full potential of machine learning, researchers have turned to RO techniques (Bertsimas and Sim, 2004) to tackle "decision-making under uncertainty". Specifically, researchers integrate RO principles into the analysis of big data and apply advanced machine learning algorithms to explore robust and reliable strategies that effectively handle uncertainties, maximise performance and achieve optimal outcomes in these dynamic environments (Bertsimas et al., 2011, 2018; Bertsimas and Sim, 2004).

RO aims to find optimal solutions that are resilient to uncertainties and variations in input parameters (Ben-Tal and Nemirovski, 2002; Uğurlu, 2024; Uğurlu and Brzeczek, 2023). The uncertainty set, the heart of RO, defines the range of possible values that the uncertain parameters can take (Ben-Tal et al., 2009; Ben-Tal and Nemirovski, 2002). Traditionally, RO methods rely on pre-defined uncertainty sets, mainly because they simplify the complexities of real-world problems. However, recent advancements in machine learning techniques offer a huge opportunity to construct more accurate data-driven uncertainty sets and thereby enhance the performance of optimisation solutions (Bertsimas and Thiele, 2006). Although statistical methods in this context have existed for several years, their integration into RO has become more important recently. This is partly due to the increased availability of large datasets, but also due to advances in several machine learning models, moving from basic statistical concepts to complex neural networks. Therefore, these developments make it more practical nowadays to construct uncertainty sets directly from data. Specifically, diverse machine learning techniques can be employed to tighten uncertainty sets which imply reducing the pessimism of robust solutions (San Juan and Sy, 2024). For this work, a machine learning integration for two-stage RO problems is studied to enhance decision-making in the presence of uncertainty. This approach accounts the dynamic nature of real-world problems by incorporating adaptability.

As a result, this chapter focuses on leveraging unsupervised learning to improve uncertainty modelling in RO. The primary objective is to propose an adaptive framework that starts from data leading to high-quality robust decisions and predictions (Bertsimas et al., 2018). Particularly, we aim to develop adaptive RO models and machine

learning algorithms that can provide solutions that are not overly sensitive to uncertainties in input parameters. This is accomplished by focusing on the uncertainty set construction. With that we aim to merge machine learning and RO, and thus improve the decision-making process by bridging the gap between a good prediction and a good decision (Ben-Tal et al., 2009; Ben-Tal and Nemirovski, 2002; Bertsimas and Sim, 2004; Bertsimas and Thiele, 2006; Birge and Louveaux, 2011; Jin et al., 2017).

Building on previous DPMM uncertainty set methods and existing ℓ_1/ℓ_∞ constructions (Campbell and How, 2015; Dai et al., 2020; Ning and You, 2019), this chapter develops a data-driven extension that integrates bootstrap resampling into the estimation of the parameters and thus enhancing the uncertainty set scaling procedure. In particular, the bootstrap estimates of component means and their variability are incorporated into the specific cluster boundaries. The contribution therefore lies in this specific integration.

Specifically, we apply Bootstrapped Dirichlet Process Mixture model to construct data-driven uncertainty sets. These uncertainty sets capture the variability present in the underlying data distributions without relying on predefined assumptions (Faccini et al., 2022; Tay et al., 2013). The unsupervised learning model learns the parameters of the Gaussian mixture components, such as means and covariances, from the data. However, achieving precise inference for the mixture parameters and weights presents an intractable challenge. Consequently, it becomes vital to identify optimal parameters and reasonably accurate coefficients within a modified uncertainty set (Gong and You, 2017). Moreover, there is a need to approximate a distribution that closely resembles the true data distribution, enabling the upcoming accurate construction of the uncertainty set (Dai et al., 2019; Ning and You, 2017). Our goal is also to refine the model such that it identifies significant patterns without being influenced by overfitting, thereby ensuring that it recognises genuine clusters in new and unseen uncertain data. Extreme or low-likelihood observations may cause the DPMM to form small and weakly supported clusters, and thereby contribute to overfitting. To reduce this sensitivity during the clustering process, the proposed framework incorporates an outlier detection mechanism (Ning and You, 2017). Therefore, those outliers are treated as a

potential contributor to overfitting, rather than its sole cause. To tackle the robustness, we employ a bootstrap resampling with replacement method for uncertainty set construction. Across the case studies, this the first time we have shown that the integration of bootstrap resampling in this specific process leads the resulting framework to improved objective values relative to the comparison methods, although the magnitude of improvement varied across experiments. Consequently, we can flexibly model the distribution of exported uncertain parameters and adjust the uncertainty sets based on observed data patterns, allowing for a robust data-driven representation of uncertainty. Our work is quite versatile and applicable to various types of data, including real-time, historical, or synthetic datasets. In domains such as production planning, where data patterns can demonstrate a range of characteristics including symmetry, asymmetry, wide spread, and typical or non-typical distributions (Garcia and You, 2015; Ning and You, 2019), our approach offers a robust solution because we approximate the best parameters based on the machine learning predictions.

3.2 Problem Statement

In this section, we formally define the research challenge in the uncertainty modelling and the nonparametric uncertainty set construction for two-stage RO. We address the challenge of constructing data-driven uncertainty sets where the decision-makers lack exact knowledge of the underlying distributions of the data and its variations. Previous conventional studies assume a predefined shape for uncertainty sets, which, however, may lead to suboptimal solutions and increased risk in decision-making processes. In addition, although a limited number of previous studies have employed DPMMs and ℓ_1/ℓ_∞ constructions to represent uncertainty, the resulting parameter estimates may remain sensitive to sampling variation. To tackle this challenge, we propose an integrated methodological extension, referred to as the Bootstrapped Dirichlet Process Mixture Model approach. Specifically, we will develop a data-driven extension that applies bootstrap resampling to the DPMM estimation and incorporates the resulting component-wise means and their variability into the uncertainty set centres and scaling parameters. The contribution therefore lies in this specific innovative integration,

which will guide the construction of the data-driven uncertainty set and thus tackle the existing problem.

Particularly, the research questions to tackle include:

RQ.1. *How to strike the right balance between optimality and conservativeness in the uncertainty set construction?*

RQ.2. *How to construct compact uncertainty sets using nonparametric information?*

RQ.3. *How to derive a method to approximate the unknown distribution, specifically the posterior distribution of latent variables?*

RQ.4. *How to further enhance the robustness of the ML parameters that influence the decisions?*

From the theoretical perspective, a simple illustration of optimisation approach is given by the objective function $J(x)$, which represents the quantity that needs to be minimised in the optimisation problem. A single data point p over the function tries to find the local or global minimum $\min_p J(p)$. Let us assume that the parameters of the model involve some kind of uncertainty, for instance $p \to p + \Delta p$, where Δp is a derivation from noise as mentioned. In this case, we do not have a single point but instead a disk of confidence interval that tries to find the best worst-case in

$$\min_{p} \max_{\Delta p} J(p + \Delta p) \tag{3.1}$$

where $\Delta p \in \mathcal{U}$ and $\mathcal{U}$ is the uncertainty set. The difference between this robust solution in contrast to nominal approach is the sensitivity to uncertainty. The nominal solution is highly sensitive to any parametric uncertainty or changes in coefficients, while the robust solution does not give such high quality solution but it is robust against these uncertainties, parameter variations and data perturbations, everything inside the Δp. As a result we have the general single stage RO formulation $\min_x \max_{u \in U} f(x, u)$ and two-stage RO, also called adaptive RO, $\min_{x,y} \max_{u \in U} f(x, y, u)$.

Consider the following two-stage RO problem:

$$\min_{x,y} \max_{u \in U} \Big(c^\top(u)\, x \; + \quad b^\top\, y(u) \Big) \tag{3.2}$$

$$s.t. \;\; A(u)\, x + B\, y(u) \leq d, \;\; u \in U$$

where $c^\top$ and A are involved in the uncertainty U, while $b^\top$ and B are not. $x \in X$ denotes the set of initial decision variables that are determined before the uncertainty u becomes known and $y \in Y(x, u)$ represents the set of secondary decisions which can be made after the uncertainty has been revealed (Bertsimas and Thiele, 2006). The initial decisions, known as "here-and-now" decisions, and the secondary decisions, termed as "wait-and-see" decisions, can both encompass continuous or discrete variables.

Overall, the primarily target in two-stage (adaptive) RO is to calculate a first-stage solution x_i such that for every uncertain scenario u_i in uncertainty set $U = \{u_1, u_2, \cdots\}$, there is a feasible second-stage solution y_i so that the worst possible objective value over all scenarios is minimised. Hence, the main challenge is to construct such precise, but also robust, uncertainty set U that will influence the solution quality.

As mentioned, in RO, the uncertainty set U is often predefined as a single convex finite set and the uncertain parameters u are expressed using a parametric distribution with a fixed set of parameters. For instance, u might have a predefined bound or follow a multivariate Gaussian distribution with fixed mean and covariance parameters.

$$\begin{aligned} U &= \{u |\; u \in R^m, \; \|u\| \leq \epsilon\} \\ U &= \{u |\; u \sim N(\mu, \Sigma)\} \end{aligned} \tag{3.3}$$

where ϵ is a predefined bound on the uncertainty, and μ and Σ are fixed parameters of the Gaussian distribution. The uncertainty set describes the uncertainty information and it is often constructed based on a priori information and simple assumptions, and thus implying overly conservative solutions (Ning and You, 2017).

The key difference between parametric and nonparametric uncertainty set construction lies in how the uncertain parameters are represented. Nonparametric uncertainty

set construction does not assume a fixed parametric form. Instead, it learns the uncertainty set directly from data, represented as a set of $\{u_i\}_{i=1}^{n}$ and from the adoption of the specific nonparametric model. Each u_i represents a realisation of the uncertain parameters, and together they form the nonparametric uncertainty set U. The nonparametric uncertainty set is represented as:

$$\begin{aligned} U_j &= \{u_i^{(j)} |\ u_i^{(j)} \in R^m,\ i = 1, \cdots, n^{(j)}\},\ \ j = 1, \cdots, k \\ U_{final} &= U_1 \cup U_2 \cup \cdots \cup U_k \end{aligned} \tag{3.4}$$

where k is the number of components of the uncertain data, $n^{(j)}$ the number of data points of that specific j component and $u_i^{(j)}$ represents the i-th realisation of the uncertain parameters assigned to component (j). It allows the number j of mixture components U_j to grow with the data, providing flexibility in modelling without specifying a fixed number of parameters k.

Using a single fixed set to describe uncertainty by fitting data and adjusting parameters is not effective in dynamic environments. As such, we aim to to derive the "optimal" polyhedral uncertainty set to capture the entire space. In addition, nonparametric machine learning models have a flexible number of parameters. Even though convex optimisation is always easier to solve, it presents these highly conservative solutions because it includes unrealistic scenarios. The idea is to combine several simple convex sets in Equation 3.4 in a non-convex way to cover the entire region and exclude unnecessary space, than previous conservative "one-set-fits-all" uncertainty sets (Bertsimas et al., 2021; Ning and You, 2017). This is recently employed by several authors, but not entirely explored due to the variety and complexity of different machine learning models and their capabilities.

In general, the overall target for compact and efficient uncertainty modelling is to strike a balance between capturing sufficient uncertainty, to account for variations in the data, while avoiding excessive shrinkage that could lead to overfitting (Dai et al., 2020). That said, our compact uncertainty set construction requires a trade-off between data coverage and conservativeness. A scale parameter that is too small may exclude important uncertainty realisations, whereas an excessively large scale may produce

overly conservative decisions. Therefore, we determine an adaptive scale parameter for each mixture component according to the coverage level α_{cov}. Its formal construction and relationship with the robust objective value are presented in the following section.

3.3 Methodology

In this section, we employ a Dirichlet process mixture model to describe the data pattern for situations where the number of clusters are unknown in the dataset. Subsequently, stick-breaking is employed to flexibly represent the uncertainty structure, through obtaining the weights of the mixture model by finding the best number of components. The final vital step is to approximate the posterior distribution of the parameters in the DPMM. Although a handful optimisation methods could be tested, Variational Inference is used to estimate the relative distributional parameters, by approximating the posterior distribution and by obtaining an approximating distribution that is computationally tractable (Chu and You, 2013; Dai et al., 2019; Yao et al., 2018). We then proceed with the heart of the research, the formulation and construction of the final data-driven uncertainty set. This includes the calculation of the conservativeness with adaptive scale parameter, reinforcing the robustness with bootstrapping and outlier detection. Additionally, we provide probabilistic guarantees for the robust data-driven uncertainty set.

3.3.1 Dirichlet process Gaussian mixture model to describe the data pattern

To accurately model a given dataset, we employ the Dirichlet process mixture model - a powerful nonparametric Bayesian statistical model. It is worth noting that despite being labelled as "nonparametric" this machine learning model does have parameters. However, the parameter space for this model is infinite-dimensional, allowing for greater flexibility in representing uncertainty (Dai et al., 2020). Infinite-dimensional parameter can significantly increase the computational cost. To address this computational issues associated with the infinite-dimensional parameter space, significant interest will be

given to variational inference which is a probabilistic modelling technique, enabling to estimate the relative distributional parameters.

Denote the data as $u = (u_1, \cdots, u_N)$. The probability distribution of u, $p(u|\theta)$, is described as:

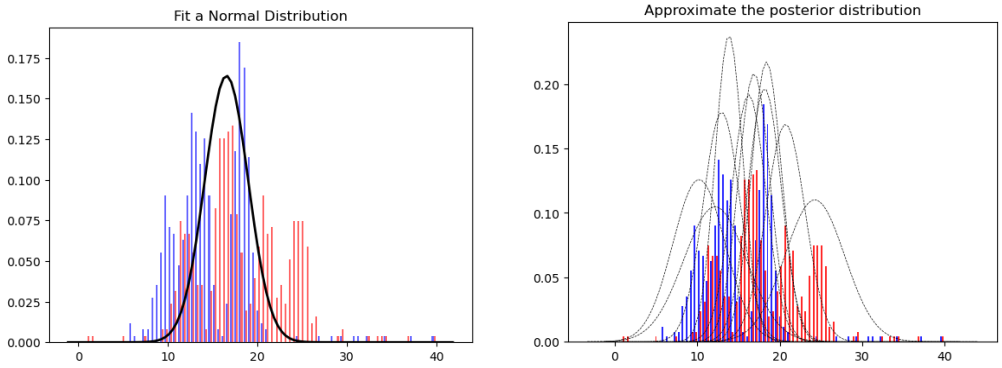
$$p(u|\pi, \theta) = \sum_{k=1}^{M} \pi_k N(u|\mu_k, \Sigma_k) \tag{3.5}$$

where $\theta = (\mu_k, \Sigma_k)_{k=1}^{M}$, $\sum_{k=1}^{M} \pi_k = 1$, $\pi_k \geq 0$ the weight for k component, $N(\cdot) \sim$ Gaussian Distribution, μ_k mean and Σ_k covariance (Campbell and How, 2015; Dai et al., 2020).

According to Dai et al. (2020) and due to the fact that the number of Gaussian components M in GMM is unknown, we require to choose the number of components M in advance in GMM. If this choice is misspecified, meaning if it is too small or too large, the fitted mixture can diverge from the true distribution. Thus if the value of M is not properly determined, there will be a significant disparity between the observed data distribution and the GMM distribution. To address this issue, we proposed the use of DPMM, which employs the Dirichlet process as the prior distribution $p(\theta)$ to determine the number of components, which is also considered as an infinite mixture model. As a result, we eliminate the previous encountered issue and the need to specify the number of components in advance, by essentially letting the data adjust the complexity of the model (Ning and You, 2019).

In Figure 3.1a we can observe that the attempt to fit a single Gaussian distribution to approximate two uncertain parameters, blue and orange, fails miserably since we do not have a accurate representation of the uncertainty with only two final exported parameters, $\theta = (\mu, \Sigma)$. However, in Figure 3.1b the resulting predicted and exported parameters $(\pi_k, \mu_k, \Sigma_k)_{k=1}^{M}$ from equation 3.5 automatically determine the number of components by considering infinite mixtures and letting the data guide the process.

The Dirichlet process is defined as follows: each uncertain parameter u_i is generated by first choosing a component that is distributed according to $\pi = (\pi_1, \cdots, \pi_M)$ and then the u_i is generated from choosing the Gaussian component with the parameter



(a) (b)

Figure 3.1: Fitting Gaussian Mixture Model(s) to find the true representation of the data.
(a) Attempting to fit a Normal Distribution to represent the data, (b) Approximate the true posterior distribution representation

θ_i, which is also generated from a prior distribution G. This follows a random draw in the Dirichlet process $DP(\alpha, G_0)$ (Campbell and How, 2015; Dai et al., 2020; Ning and You, 2019). Essentially, we place a Dirichlet–process prior on the unknown mixing distribution G. For each observation k, we draw a latent parameter $\theta_k \sim G$ and we then sample u_k, as shown below:

$$\begin{aligned} \theta &\sim G \\ G &\sim DP(\alpha, G_0) \\ u_k &\sim N(u|\mu_k, \Sigma_k) \end{aligned} \tag{3.6}$$

where $\alpha \sim Gamma(a_0, b_0)$ to avoid the direct specification of the concentration parameter α as mentioned in Campbell and How (2015) and G_0 is the base distribution. The concentration parameter α has a tremendous effect on the number of components within the data (Blei and Jordan, 2006). Lastly, the mean μ_k and the covariance Σ_k are unknown and we have to set the base distribution G_0 as a Normal–Wishart distribution:

$$G_0 \sim N(\mu|\mu_0, (\beta_0\Lambda)^{-1})\, W_k(\Lambda|\Psi_0, \upsilon_0) \tag{3.7}$$

where $W_k(\cdot)$ denotes the Wishart distribution, Λ denotes the precision and μ_0, β_0, Ψ_0, υ_0 are the fixed hyperparameters of the base distribution G_0.

Cluster allocation with stick-breaking

Following Dai et al. (2020) thought process, the distinct values of $\theta = (\theta_1, \cdots, \theta_n)$ lead to the division of the dataset into clusters, where each cluster k consists of elements with the same value θ_k^*, and the distribution that describes these divisions is known as the Chinese restaurant process. Intuitively, the process favours joining well-populated clusters while still allowing the appearance of genuinely new clusters. Specifically, a dataset is divided into an infinite number of clusters, and each cluster can accommodate an infinite number of data points (Caron et al., 2007). As a sequence of $\{\theta\}_{k=1}^{N-1} = (\theta_1, \cdots, \theta_{N-1})$ is observed, each value is assigned to a cluster and the N^{th} value, θ_N, has the option to either join an existing cluster or form a new one. The concentration parameter α controls this trade-off, with larger values encouraging more and smaller clusters and smaller values encouraging fewer and larger clusters. This process is provided and described in the following form of distribution:

$$\theta_N = \begin{cases} \frac{N_c}{\alpha+N-1}\ \delta(\theta_N = \theta_k^*), & \text{for an existing cluster} \\ \frac{\alpha}{\alpha+N-1}\ \theta, \ \ \theta \sim G_0, & \text{for a new cluster} \end{cases} \tag{3.8}$$

where N_c represents the total number of data points assigned to cluster k (i.e., the total number of samples that have a value equal to θ_k^*) and δ is the delta function. It can be observed in Equation 3.8 that the posterior distribution when assigning a value θ_N: there is a probability of $\frac{\alpha}{\alpha+N-1}$ for it to be allocated to a new cluster, or a probability of $\frac{N_c}{\alpha+N-1}$ to be assigned to an existing cluster (Dai et al., 2020). This means that parameter sets should be automatically grouped into discrete clusters, with adjustments made based on the samples allocated to each cluster.

As a result, the stick-breaking construction is a fundamental concept in Bayesian nonparametric modelling for our cluster allocation (Campbell and How, 2015; Caron et al., 2007). It involves breaking a unit-length stick into an infinite number of pieces, each representing the weight assigned to the above potential cluster. The key idea

behind stick-breaking is to iteratively break off portions of the stick V_k according to a sequence of probabilities, which are determined by a beta distribution. The stick-breaking construction and then the two collections of random variables are provided as follows:

$$\begin{aligned} G &= \sum_{k=1}^{\infty} \pi_k(V)\delta(\theta = \theta_k^*) \\ V_k &\sim Beta(1, \alpha) \\ \theta_k &\sim G_0 \end{aligned} \tag{3.9}$$

where the generation of the weight $\pi_k(V) = V_k \prod_{j=1}^{k-1}(1 - V_j)$. Initially, to explain this process, we assume to have a stick of length 1, then we break at V_1 with corresponded weight of π_1 which is the length of the broken-off portion. Subsequently, we repeatedly break the remaining portion to generate a full set of $\{\pi_k : k = 1, 2, \cdots\}$. Simultaneously, for each broken-off portion of the stick, a corresponding cluster is created, represented by a unique parameter value drawn from a base distribution G_0. This allocates weights in an effective manner to an infinite number of potential clusters, with the weights determined by the stick lengths and the clusters defined by the drawn parameter values. The function $\delta(\theta = \theta_k^*)$ in Equation 3.9 represents an indicator function centred on the optimal θ_k^* (Dai et al., 2020).

Optimising posterior approximations with variational inference

To approximate the posterior distribution of latent variables we will use Variational Inference (VI), an optimisation-based approach applied to fit probabilistic models. While Variational Inference involves optimisation, it does not explicitly have a step size parameter or a learning rate such as gradient-based optimisation algorithms. Instead, we have parameters that control the optimisation process, such as the number of components in the data, the concentration parameter and other associated hyperparameters. We do not know the actual posterior distribution of the data, thus the target is to approximate this posterior distribution through obtaining an approximating distribution, a surrogate, that is computationally tractable and as good as possible. This will

be done by introducing the KL Divergence, which will be the distance between the distributions $p(\cdot)$ and $q(\cdot)$. The overall aim is to minimise this divergence between the true posterior distribution and the approximating (variational) distribution (Blei and Jordan, 2006).

The surrogate posterior distribution is given below:

$$q(\alpha, Z, V, \mu, \Sigma) = q_1(\alpha) \prod_{i=1}^{N} q_2(Z_i) \prod_{k=1}^{M-1} q_3(V_k) \prod_{k=1}^{M} q_4(\mu_k, \Sigma_k) \tag{3.10}$$

where $q_1(\alpha)$ denotes the previously mentioned Gamma distribution in Equation 3.6, $q_2(Z_i = k) = \frac{\rho_{ik}}{\sum_{j=1}^{M} \rho_{ij}}$, where ρ is the not-yet normalised probability component for data point i and cluster k (Ning and You, 2017), $q_3(V_k)$ the Beta distribution and $q_4(\mu_k, \Sigma_k)$ the Normal-Wishart distribution from equation 3.7 (Campbell and How, 2015; Ning and You, 2017).

We define U as the observations, the observed data. The (Z, V, μ, Σ) are the latent and hidden variables, while α is the hyperparameter. The Dirichlet process mixture model from the previous subsection defines the joint distribution $p(U, \alpha, Z, V, \mu, \Sigma)$ that we know of both the observed data U and the parameters $(\alpha, Z, V, \mu, \Sigma)$. The posterior distribution combines the prior distribution $p(\alpha, Z, V, \mu, \Sigma)$ with the likelihood of the observed data $p(U \mid \alpha, Z, V, \mu, \Sigma)$ to update those beliefs, divided by the marginal $p(U)$. The overall goal of that inference is to find the posterior distribution of the latent variables and parameters $z = (\alpha, Z, V, \mu, \Sigma)$ given the observation uncertain parameters U, expressed as $p(\alpha, Z, V, \mu, \Sigma \mid U) = p(z \mid U) = p(U \mid z) * p(z)/p(U)$. This represents the true posterior distribution and the updated belief after observing the data, combining the prior and the likelihood. By "inference" we mean that we want to use our knowledge on observed data in order to find information on the latent variable. The problem in finding the true posterior distribution is finding or evaluating the marginal $p(U)$, which is intractable.

The solution is to use a surrogate $q(\cdot)$ which will be almost as good as possible with the true posterior distribution, $q(z) \approx p(z \mid U)$. Thus $q(z) = q(\alpha, Z, V, \mu, \Sigma)$ is the surrogate or variational distribution (Campbell and How, 2015) we are trying to

approximate. By "variational" we mean optimise over functions. Finally, the parameters of each term in Equation 3.10 are estimated by minimising a loss metric, which will be the KL divergence between the true posterior and this approximated variational distribution (Campbell and How, 2015):

$$\arg\min_{q\in\mathcal{Q}} \mathrm{KL}\Big(q(\alpha, Z, V, \mu, \Sigma) \,\|\, p(\alpha, Z, V, \mu, \Sigma \mid U)\Big) = \min E_{z\sim q(z)} \left(\log \frac{q(\alpha, Z, V, \mu, \Sigma)}{p(\alpha, Z, V, \mu, \Sigma \mid U)}\right) \tag{3.11}$$

However, we do not have access to the posterior $p(\alpha, Z, V, \mu, \Sigma \mid U)$. We only have the joint $p(U, \alpha, Z, V, \mu, \Sigma)$. We need to reformulate as follows:

$$\begin{aligned}
\mathrm{KL}\Big(q(z) \,\|\, p(z \mid U)\Big) &= \mathbb{E}_{z\sim q(z)}\left[\log \frac{q(z)}{p(z \mid U)}\right] \\
&= \int_{\mathcal{Z}} q(z)\, \log\left(\frac{q(z)}{p(z \mid U)}\right) \mathrm{d}z \\
&= \int_{\mathcal{Z}} q(z)\, \log\left(\frac{q(z)\, p(U)}{p(z, U)}\right) \mathrm{d}z \\
&= \int_{\mathcal{Z}} q(z)\, \log\left(\frac{q(z)}{p(z, U)}\right) \mathrm{d}z \;+\; \int_{\mathcal{Z}} q(z)\, \log p(U)\, \mathrm{d}z \\
&= \mathbb{E}_{z\sim q(z)}\left[\log\left(\frac{q(z)}{p(z, U)}\right)\right] + \mathbb{E}_{z\sim q(z)}[\log p(U)] \\
&= -\,\mathbb{E}_{z\sim q(z)}\left[\log\left(\frac{p(z, U)}{q(z)}\right)\right] + \log p(U) \\
\therefore \mathrm{KL}(\mathrm{q} \,\|\, \mathrm{p}) &= -\mathrm{ELBO}(\mathrm{q}) + \log p(U)
\end{aligned} \tag{3.12}$$

We need to mention that from the exported equation in Equation 3.12, the $p(U)$ is the marginal which is a probability between 0 and 1, and if we apply the $\log p(U)$ we get the evidence which is something negative (≤ 0) and fixed. On the other hand, KL is a distance thus it is something positive and non-zero (> 0). Consequently, the ELBO should be negative (≤ 0), but also smaller than the evidence $\log p(U)$, thus we call it evidence lower bound (ELBO). As a result, our initial aim to find the $\arg\min_{q\in\mathcal{Q}} \mathrm{KL}$ is equivalent to find the $\arg\max_{q\in\mathcal{Q}} \mathrm{ELBO}$. The process of VI is detailed in Algorithm 1,

while "converge" implies the stopping rule when we reach an ELBO stabilisation, like Ning and You (2017).

Algorithm 1 Variational Inference Algorithm

Input uncertainty data set u_i and model $p(\cdot)$
Initialize Variational factors $q(\alpha, Z, V, \mu, \Sigma)$ in eq.3.10
while $ELBO \not\to$ converge **do**
    Update $q_1(\alpha)$
    for i = 1,$\cdots$,N **do**
        Update $q_2(Z_i)$ in eq.3.10
    end for
    for k = 1,$\cdots$,M **do**
        Update $q_3(V_k)$ in eq.3.10
        Update $q_4(\mu_k, \Sigma_k)$ in eq.3.10
    end for
    Compute $ELBO = E\{\log p(U, \alpha, Z, V, \mu, \Sigma)\} - E\{\log q(\alpha, Z, V, \mu, \Sigma)\}$
end while
Result: Variational Mean-Field Approximation $q(\alpha, Z, V, \mu, \Sigma)$ in eq.3.10

Figure 3.2 elaborates the Variational Inference process. This iterative process continues until convergence is reached. Stick-breaking allows for automatic determination of the number of clusters based on the available data. For the DPMM, we are able to estimate the optimal number of components, M, along with the exported parameters, $(\alpha, Z, V, \mu, \Sigma)$. Finally, we will then employ in the next subsection the useful approximated parameters, $(\pi_k, \mu_k, \Sigma_k)_{k=1}^{M}$, for the construction of the data-driven uncertainty set.

3.3.2 Construction of the data-driven uncertainty set

The target is to construct a compact uncertainty set U directly from data that captures the realisation of uncertain parameters in u, in Equation 3.4. This computational tractable data-driven uncertainty set is constructed based on the previous approximated parameters of the distribution in Equation 3.10. We iteratively solve the Algorithm 1, initializing the prior, running the variational inference, and exporting a set of optimal parameters $\{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{M}$. Based on these extracted GMM components and parameter estimates, the uncertainty set is constructed as a combination of multiple diverse uncertainty sets. Mathematically, the proposed union of data-driven uncertainty sets

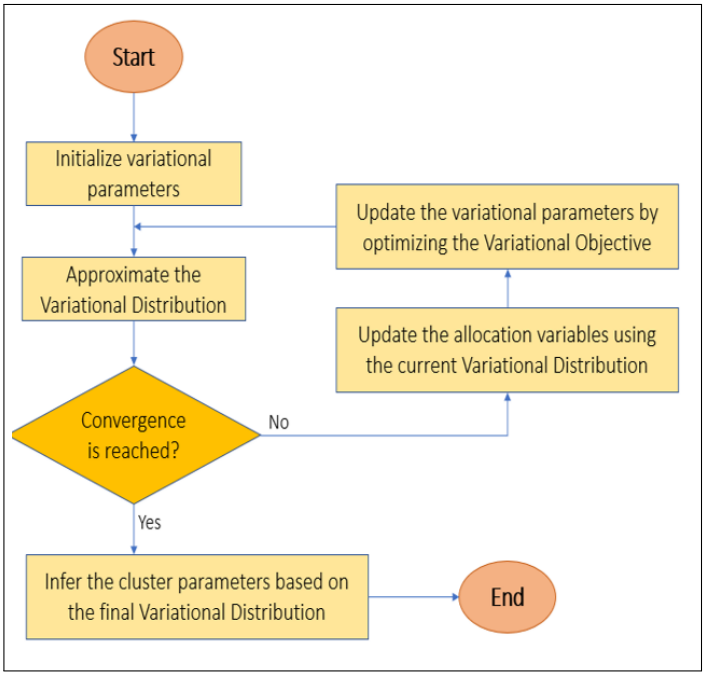


Figure 3.2: The process for Variational Inference in DPMM

can be expressed as follows:

$$\mathcal{U} = \bigcup_{\pi_k \geq \pi^*} \Big(u |\ \|\tau_k(u-\mu_k)\|_1 \leq (1+\sigma_{\mu_k})\Gamma_{1,k}^{\alpha_{cov}}, \\ \|\tau_k(u-\mu_k)\|_\infty \leq (1+\sigma_{\mu_k})\Gamma_{\infty,k}^{\alpha_{cov}} \Big) \tag{3.13}$$

where u is the uncertainty introduced. The threshold π^* is usually set to 5% (Dai et al., 2020; Ning and You, 2017), while π_k is the weight of the k component for $k = 1, 2, ..., M$, meaning the probability of that k mixture component. It is calculated from Algorithm 1 and $q_3(V_k)$ of equation 3.10, where $q_3(V_k) = Beta(\eta_k, \nu_k)$. Kindly note that η_k and ν_k are the hyper-parameters for Beta distribution in the Variational inference, thus $\pi_k = \frac{\eta_k}{\eta_k+\nu_k}\prod_{j=1}^{k-1}\frac{\nu_j}{\eta_j+\nu_j}$, for $k = 1, \cdots, M-1$ and $\pi_M = 1 - \sum_{k=1}^{M-1}\pi_k$ (Campbell and How, 2015; Ning and You, 2017). Components with a weight π_k greater than the threshold π^* are included in the uncertainty set. The coverage rate α_{cov} is a predetermined value representing the desired extent of the uncertainty set, for example $\alpha_{cov} = 1$ means 100% coverage rate. This solely influences the adaptive scale parameters

$\Gamma_{1,k}^{\alpha_{cov}}$ and $\Gamma_{\infty,k}^{\alpha_{cov}}$, which are associated with each cluster and determine the structure and size. In detail, the coverage rate α_{cov} is set in advance according to the predefined data coverage rate the user prefers, how conservative or flexible the user wants to be. For simplicity, we will proceed with the full data coverage rate in this study and it aligns well with (Dai et al., 2020), where the authors chose 99.99% predefined data coverage rate. Thus, we will choose to include all relevant (non-outlier) data points within the uncertainty range, while we already set 95% coverage in the distribution before to focus on the most significant uncertainty. μ_k the mean and σ_{μ_k} the standard deviation of the mean belong to the bootstrap resampling step in the following subsection. σ_{μ_k} slightly expand this boundary to account for small perturbations or variations in the data.

Our construction is based on the classical uncertainty $U = \{z| \ \|\frac{(z-z_0)}{\sigma}\|_p \leq \rho\}$ (Bertsimas et al., 2011, 2021, 2018; Bertsimas and Kim, 2024). For our uncertainty set construction, τ_k will have a new role and will be an adaptive parameter that changes depending on the type, structure, and variability of each component around the mean μ_k. We assign it as $\tau_k = \Sigma_k^{-1}$, which is the value of the corresponding inverse covariance matrix for each component, or as $\tau_k = (L_k^{-1})^T$, which is the Cholesky decomposition of Σ_k^{-1}, where $\Sigma^{-1} = (LL^T)^{-1} = (L^T)^{-1}L^{-1}$. This is because in some cases the cluster data points are spread in an unbalanced way around the μ_k and there is uneven variability. We need a more accurate and adaptive manner to better capture the structure of the cluster, as well as less conservative bounds compared to fixed uncertainty sets, leading to better clustering performance. The uncertainty set construction is inspired by Dai et al. (2020), where the authors used a similar way to fix the orientation of their components but utilised the upper triangular matrix derived using the Cholesky decomposition of the result of the inverse covariance inference for each component. This representation is especially useful for estimating distances between observations in these Gaussian distributions. However, the authors did not integrate the intersection of the norms into a holistic uncertainty set. Specifically, they formulated the ℓ_1 and ℓ_∞ constructions as two alternative uncertainty sets and evaluate them separately, without providing a rationale for not intersecting them. In this study, both constraints are instead imposed simultaneously for each estimated component. In general, as pre-

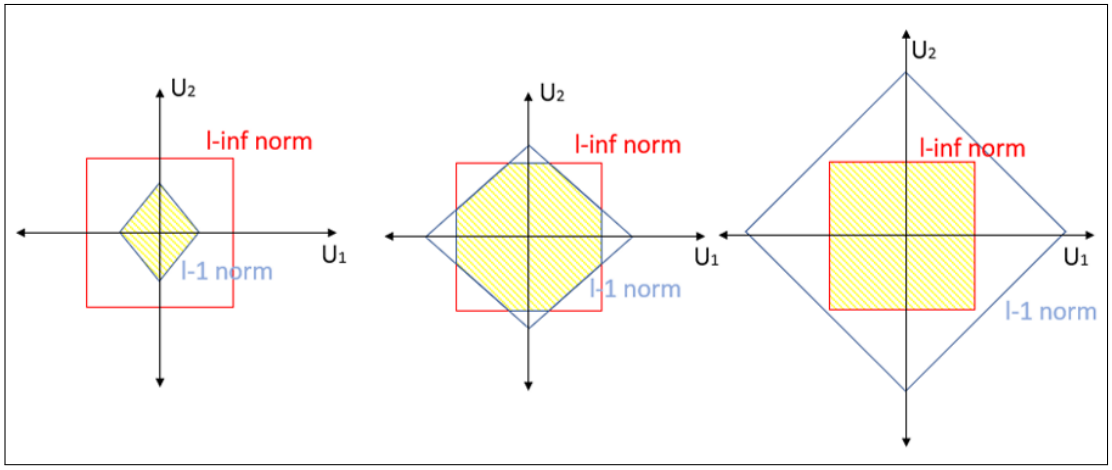


Figure 3.3: Intersection of l_1 and l_∞ norms

sented in Figure 3.3, for each component/cluster, we construct the intersection of: $\{\|\tau_k(u-\mu_k)\|_1 \leq (1+\sigma_{\mu_k})\Gamma_{1,k}^{\alpha_{cov}}\}$ and $\{\|\tau_k(u-\mu_k)\|_\infty \leq (1+\sigma_{\mu_k})\Gamma_{\infty,k}^{\alpha_{cov}}\}$. Subsequently, we take the union of all the constructed data-driven uncertainty sets. Our framework is a combination of both probabilistic and worst-case approaches.

The union of intersected uncertainty sets results in significant challenges, and thus we divide the problem into manageable subproblems. Particularly, we employ the master-subproblem framework combined with Column-and-Constraint Generation to solve the optimisation problem. This approach is used by several authors (Dai et al., 2020; Goerigk and Kurtz, 2022; Ning and You, 2017), because it is easy to implement and it is particularly effective due to its ability to handle non-convexity.

Calculating the conservativeness

The level of conservativeness in the uncertainty set is determined by the scale parameters $\Gamma_{1,k}^{\alpha_{cov}}$ and $\Gamma_{\infty,k}^{\alpha_{cov}}$. However, when dealing with the U uncertainty set, there is no straightforward method to compute the scale parameter and therefore we need to choose the best hyperparameter Γ for U. To address this, we have propose a modified methodology from Dai et al. (2020) to determine the minimum scale parameter that achieves the desired data coverage rate. In this proposed method, we essentially provide the mean value μ_k and τ_k of each component k into the equation 3.14. This will

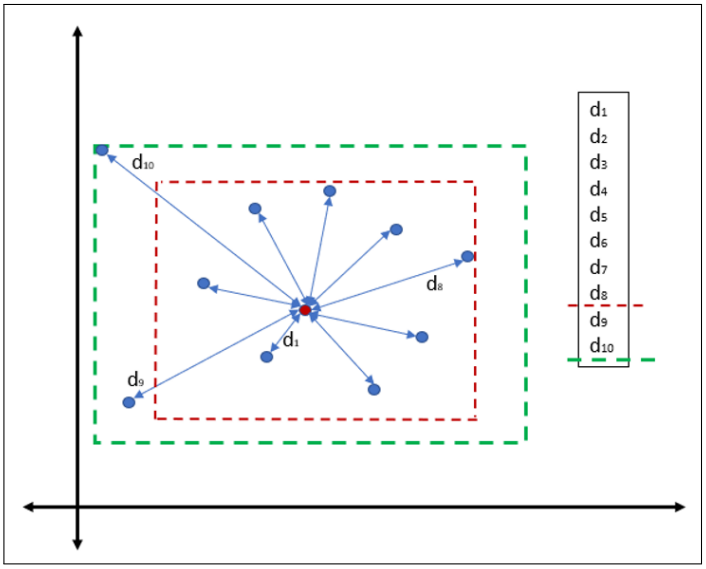


Figure 3.4: Calculation of Gamma for l_∞ norm

result to the estimation of ξ_i for each data point u_i in U:

$$\xi_i = \|\tau_k(u_i - \mu_k)\| \tag{3.14}$$

where τ_k is defined as above and the $\|\cdot\|$ norm changes depending the norm we are calculating, $\|\cdot\|_1$ for Γ_1 and $\|\cdot\|_\infty$ for Γ_∞. For the desired data coverage rate α_{cov}, $\Gamma_{1,k}^{\alpha_{cov}}$ and $\Gamma_{\infty,k}^{\alpha_{cov}}$ are considered as the N^{th} smallest ξ_i, as shown in Equation 3.15:

$$N = round(N_k \times \alpha_{cov}) \tag{3.15}$$

where $round(x)$ is a function that returns the closest integer to x. For each component k, N_k is the number of total data points.

For illustration, we show the calculation process in Figure 3.4 where 10 data points with each representing 1 identified component. Red point is considered the mean μ and blue points are the data points u_i. $\xi_i = \|(u_i - \mu)\|$ calculates the distance of each point from its mean. We get a sorted array with the distance for each ξ_i. For $\alpha_{cov} = 1$, we apply these to the $N = round(N_k \times \alpha_{cov}) = 10$. So we select the 10th smallest ξ from our sorted array which is the d_{10}. As can be observed, selecting $\xi_{10} = d_{10}$ as our Γ_∞ for

that component, we include all the data points in our green dashed box lines. If we select $\alpha_{cov} = 0.8$ coverage rate, we apply these to the $N = round(N_k \times \alpha_{cov}) = 8$. Hence we select the 8th smallest ξ, thus the distance with d_8 as our Γ_∞ for that component, observing that we exclude 2 data points from the red dashed line box. The same will be applied for Γ_1 for that component and the same for every predicted component.

More formally, we let

$$\xi^{(p)}_{(1),k} \leq \cdots \leq \xi^{(p)}_{(N_k),k}, \qquad p \in \{1, \infty\},$$

denote the values of $\xi^{(p)}_{i,k} = \|\tau_k(u_i - \mu_k)\|_p$ from Equation 3.14. For the above desired data coverage rate α_{cov} and from Equation 3.15 with $1 \leq N \leq N_k$, we define

$$\Gamma^{\alpha_{\text{cov}}}_{p,k} = \xi^{(p)}_{(N),k}$$

Holding the fitted parameters and components fixed, if $0 < \alpha_{\text{cov},1} \leq \alpha_{\text{cov},2} \leq 1$, then, for $p \in \{1, \infty\}$,

$$\Gamma^{\alpha_{\text{cov},1}}_{p,k} \leq \Gamma^{\alpha_{\text{cov},2}}_{p,k}, \qquad \mathcal{U}(\alpha_{\text{cov},1}) \subseteq \mathcal{U}(\alpha_{\text{cov},2})$$

Therefore, defining the robust optimal objective value as

$$z^\star(\alpha_{\text{cov}}) = \min_{x \in \mathcal{X},\, y(\cdot)} \max_{u \in \mathcal{U}(\alpha_{\text{cov}})} \Big(c^\top(u)x + b^\top y(u) \Big)$$

$$\text{s.t.} \quad A(u)x + By(u) \leq d, \qquad \forall u \in \mathcal{U}(\alpha_{\text{cov}})$$

which is the two-stage RO problem in Equation 3.2 under $\mathcal{U}(\alpha_{\text{cov}})$, we obtain

$$z^\star(\alpha_{\text{cov},1}) \leq z^\star(\alpha_{\text{cov},2})$$

As a result, increasing the coverage level enlarges the uncertainty set and may increase the conservativeness of the robust solution, whereas decreasing it may improve, or leave unchanged, the objective value at the cost of lower coverage.

Bootstrapped resampling method

To tackle the trade-off between optimality and conservativeness, we establish a bootstrapping method to further enhance the robustness, which, to the best of our knowledge, has not been attempted yet. We employ bootstrap resampling as a robust statistical technique to estimate the variability of the exported parameters $\{\mu_k, \Sigma_k\}_{k=1}^{M}$ of the DPMM for each k cluster and M the predicted maximum number of clusters. Our bootstrap approach involves randomly sampling n observations, size of the dataset, from the observed dataset X with replacement to create bootstrap samples X^*. Repeating this process $L = 1000$ times, we obtain a collection of bootstrap samples $X_1^*, X_2^*, \cdots, X_L^*$ upon which we apply our DPMM to each single subset to obtain the estimated mean $\mu_{X_j^*}$ $\forall 1 \leq j \leq L$. Subsequently, we take the mean of all the estimated means resulting the final mean all the collection of bootstrap samples, μ^*, which we will replace instead of the normal non-bootstrapped exported μ in Equation 3.13 and equation 3.14. For simplicity, in those equation we will refer them as μ but they are actually the bootstrapped mean μ^*. In Figure 3.5 we present the histogram of the means for each uncertain parameters, the mean of those mean, observing the standard deviation and presenting the final cluster behaviour of a single cluster mean in contrast to normal DPMM. Furthermore, the confidence level and standard deviation σ_{μ^*} of the estimated final bootstrapped mean is added as a percentage to Γ to enhance the robustness by accounting the actual deviation of the bootstrapped parameters. This implies that we do not solely rely on the calculation of the conservativeness from above method in Equation 3.14, but we also further enhance the robustness of our statistical inference (Bertsimas and Van Parys, 2022).

The algorithm of the bootstrapping resampling is presented in Algorithm 2. Specifically, we calculate the standard deviation of the bootstrapped resampled means, $\sigma_{\mu_k^*}$ for each k cluster, and add it algebraically to the Γ_k in Equation 3.13 which slightly expand the boundary to account for any potential small variation in the data. This standard error represents the single standard deviation of the bootstrap mean and holds the variability associated with the estimate of the mean obtained through the bootstrapping process. $(1 + \sigma_{\mu_k})\Gamma_{\infty,k}^{\alpha_{cov}}$ provides us a controlled expansion of the modified

uncertainty set, ensuring that it is neither too conservative nor too restrictive, without overestimating the uncertainty.

Algorithm 2 Bootstrap Procedure

Initialize X dataset and $StoredParameters = \emptyset$
for j = 1,...,L **do**
    Take n random observations from X with replacement
    Create collection of bootstrap estimation X_j^*
    Solve the DPMM with X_j^* data
    Export estimated parameter $\mu_{X_j^*}$
    Store $\mu_{X_j^*} \rightarrow StoredParameters$
end for
Calculate the mean of the $StoredParameters$, μ^*
Return: μ^* and σ_{μ^*}

However, as expected, this introduction and extra step of bootstrapped resampling increases the computational cost relative to the non-bootstrapped DPMM approach. This is due to the fact that it replaces the component means estimated from the original dataset with bootstrap averaged estimates and incorporates their estimated variability into the specific uncertainty set of that component, as described above. Consequently, this is important because it changes the location and geometry of the uncertainty set, which may alter the worst-case realisations and, therefore, the resulting first-stage and second-stage decisions and objective value. We may argue that the parameters already have well-defined confidence intervals in the data distribution, thus additional bootstrapping may be redundant and leads to this extra computational effort. Nevertheless, the second layer has a distinct role rather than reproducing these existing intervals. The DPMM provides the parameter estimates on the observed dataset, whereas the additional bootstrap repeatedly resamples the dataset and refits the DPMM to measure the sensitivity of the exported estimates to sampling variation. In particular, we incorporate this variability into the centres and boundaries of the uncertainty sets, thereby accounting the estimation uncertainty into the RO model. As a result, for the exported parameter μ, the mean location of the uncertainty set will shift to represent a more generalised centre of the data and will ensure that the uncertainty set does not rely on one specific data sample or on a single DPMM fit.

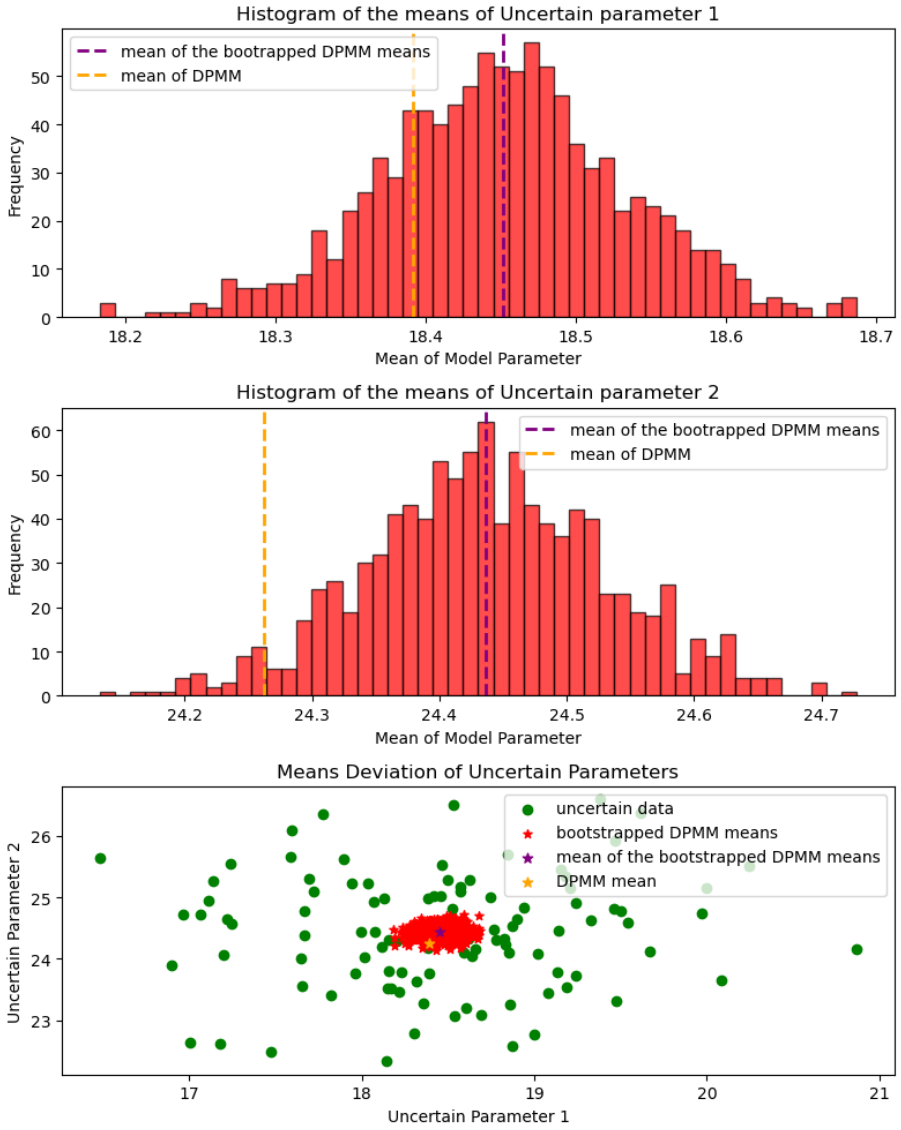


Figure 3.5: Bootstrapping method with 1000 iteration resampling for a single cluster. How to calculate the means of bootstrapped μ for one single cluster and the μ_k^* of that k cluster. Comparison is also made with DPMM mean without bootstrapped resampling.

To summarize, we propose a two-layer bootstrap framework to construct more reliable uncertainty sets and address uncertainty from different directions and perspectives. The first layer of bootstrapping within the DPMM, which estimates distributions over mixture components using resampling. The second layer involves the introduction of statistical bootstrap, essentially bootstrap of bootstrapping, which further updates the uncertainty set. The reason we employ this additional layer of resampling is to further reinforce the robustness, better capture the variability and obtain a more robust clustering solution, at the expense of slightly higher computational cost.

Outlier detection

Here, we present the procedure on how our framework identifies anomalous data points in our dataset. Particularly, our approach utilises the likelihood values from the previous DPMM to evaluate each data point. In this variational inference step, the model identifies which cluster each data point belongs to and, specifically, the main target is to identify data points with low likelihood values which implies potential outliers. The outlier detection algorithm is summarised in Algorithm 3 and provides a framework for identifying and isolating anomalous data points based on their likelihood values. If a data point has a low posterior probability of belonging to any cluster, it is considered an outlier. This is determined by the *threshold*, usually 5%, to decide whether a point represents measurement error in our dataset, thus an outlier. This will also help us in terms of generalisation and robustness on unseen data, since any future data points that form new clusters with very few members are considered outliers. Our method is quite similar to the inner-outer layers system from Ning and You (2017), which provides us robustness to any potential variation in the resulting *CleanedDataset*.

In this study, the 5% threshold is adopted as a practical criterion for identifying observations that are weakly supported by the fitted DPMM and are therefore treated as potentially anomalous or contaminated data. The purpose of this filtering step is to prevent isolated observations from disproportionately enlarging the uncertainty set and consequently producing overly conservative robust solutions. Nevertheless, a low-likelihood observation may in some cases correspond to a genuine rare adverse real-

isation, rather than contamination. Therefore, removing such observations introduces a trade-off between limiting excessive conservatism and keeping protection against rare events. As described above, this risk may be partially mitigated by the subsequent bootstrap enlargement of the uncertainty set boundaries through the factor $(1 + \sigma_{\mu_k})$ in Equation 3.13, which accounts for variability. However, we acknowledge that this enlargement does not guarantee the inclusion of every observation wrongly identified as an outlier. This is a persistent challenge in the field with similar works from Ning and You (2017) and Dai et al. (2020).

Algorithm 3 Outlier Detection method of DPMM

Fit DPMM to the data c ∈ X
Initialize $CleanedDataset = \{c_1, c_2, \cdots, c_N\}$
Initialize $OutlierDataset = \emptyset$
Calculate the log probability of each data point LogProb(c_i) using the relevant score samples function
We set *threshold*
for i = 1,...,N **do**
    if LogProb(c_i) $\leq$ *thresshold* **then**
        $c_i \rightarrow Outlier$
        Remove c_i from $CleanedDataset$
        Store c_i to $OutlierDataset$
    end if
end for
Return: $CleanedDataset$

3.3.3 Probabilistic guarantees of the proposed method

The following outlines the probabilistic guarantees and its importance to show the robustness of our approach. The algorithms presented—Algorithm 1, Algorithm 2 and Algorithm 3—are justified by their role to estimate the important parameters for constructing the final data-driven uncertainty set. In general, this uncertainty set comes with a predetermined level of probability, which shows the likelihood that the true parameter values will fall within the proposed uncertainty set. The robust complexity, $\rho(U)$, will be used to compute how hard and how difficult we satisfy a robust constraint given the uncertainty set U.

In the outlier detection process of Algorithm 3, we create an uncertainty set which

is free from data points that would lead to extreme results. We will now provide the theoretical properties of the intersection and union of the uncertainty sets, because we want to ensure that we are working with reliable uncertainty sets. Particularly, within this context of data-driven uncertainty sets, the concept of intersection between the l_1 and l_∞ norms is explored. The intersection $l_1 \cap l_\infty$ set and uncertainty $u \in R^L$ is defined as:

$$U_{l_\infty \cap l_1} = \{u : \ \|u\|_\infty \leq 1, \ \|u\|_1 \leq \rho\} \tag{3.16}$$

Based on the previous work from Bertsimas et al. (2021), we have the following:

Proposition 1 *For two uncertainty sets, U_1 and U_2 such that $ri(U_1) \cap ri(U_2) \neq \emptyset$, the robust complexity of $U_1 \cap U_2$ satisfies $\rho(U_1 \cap U_2) \geq \min_{i=1,2} \rho(U_i)$.*

Denote δ^* as the support function of U and the relative interior of U, $ri(U) := \{x \in U : \forall y \in U \quad \exists \lambda > 1, \quad y + \lambda(x - y) \in U\}$, where U is any non-empty convex set. If $U = U_1 \cap U_2$ with $ri(U_1) \cap ri(U_2) \neq \emptyset$, we have $\delta^*(y|U) = \min_\nu(\delta^*(\nu|U_1) + \delta^*(y - \nu|U_2))$. By definition of the robust complexity $\delta^*(\cdot|U_i) \geq \rho(U_i)\| \cdot \|_2$, it follows:

$$\delta^*(y|U) \geq \min_{i=1,2} \rho(U_i) \min_\nu \{\|\nu\|_2 + \|y - \nu\|_2\} \geq \min_{i=1,2} \rho(U_i)\|y\|_2$$

To generate a flexible uncertainty set construction, we use the unique characteristics of l_1 and l_∞ norms. The union of the norms, $U_1 \cup U_2, \cdots$ offers a comprehensive approach to include a range of uncertainties. For $l_1 + l_\infty$ set and $u \in R^L$, define:

$$U_{l_\infty + l_1} = \{u = u_1 + u_2 : \ \|u_1\|_\infty \leq \rho_1, \ \|u_2\|_1 \leq \rho_2\} \tag{3.17}$$

Proposition 2 *For two uncertainty sets, U_1 and U_2, the robust complexity of the Minkowski sum $U_1 + U_2$ satisfies $\rho(U_1 + U_2) \geq \rho(U_1) + \rho(U_2)$.*

Given $U = U_1 + U_2$, it follows $\delta^*(y|U) = \delta^*(y|U_1) + \delta^*(y|U_2)$. By definition of the

robust complexity $\delta^*(\cdot|U_i) \geq \rho(U_i)\|\cdot\|_2$, we have:

$$\delta^*(y|U) \geq \rho(U_1)\|y\|_2 + \rho(U_2)\|y\|_2.$$

Taking the minimum over all y such that $\|y\|_2 = 1$ proves the above.

Algorithm 1 generates the parameters for the uncertainty set construction, which are then further reinforced their robustness and generate more robust estimates with Algorithm 2. Furthermore, we identify the actual mixture components that contribute to this process. Propositions 1 and 2 extend to any finite collection of sets under the same assumptions, while the union of multiple sets is addressed in the subsequent proposition. Specifically, we show below how the union of these exported uncertainty sets, which each represent a mixture component, contributes to robust constraints and that the final uncertainty set U provides probabilistic guarantees.

Based on the previous work from Campbell and How (2015), we have the following:

Proposition 3 *For any robust constraint $c^\top x \leq b$, $\forall c \in U$, if the uncertainty set U can be expressed as a union*
$U(\epsilon) = \cup_{k=1}^{M} U_k$, then the robust constraint is equivalent to $c^\top x \leq b$ $\forall c \in U_k$, $k = 1, \ldots, M$.

Let the uncertainty set U be expressed as a union of subsets $U(\epsilon) = \cup_{k=1}^{M} U_k$. For any $c \in U$, there exists at least one subset U_k such that $c \in U_k$. Therefore, the constraint $c^\top x \leq b$ must hold for all $c \in U_k$ for each $k = 1, \ldots, M$. Hence, the original robust constraint $c^\top x \leq b \ \forall \ c \in U$ is equivalent to $c^\top x \leq b \ \forall \ c \in U_k, \ \forall \ U_k$ (Campbell and How, 2015).

Proposition 4 *The probability of feasibility of the constraint $P(c^\top x \leq b)$ is bounded below ($\geq$) by $(1-\epsilon)\sum_{k:\pi^k \geq \pi_*} \pi_k$.*

Considering the probability of feasibility, denote $\Gamma \geq 0$ as the adaptive scale parameter that satisfies $P(\|\xi\| \leq \Gamma_k) \geq 1 - \epsilon$ and $\xi \sim t_n(0, I)$, following a Student's t-distribution

with n degrees of freedom, the overall probability of feasibility is the weighted sum of the probabilities over the subsets. Hence, it is bounded below by $1-\epsilon \geq (1-\epsilon)\sum_{k:\pi_k \geq \pi^*} \pi_k$ (Campbell and How, 2015).

Therefore, the probabilistic guarantees established above apply to the corresponding uncertainty set construction under the stated assumptions. Also, the subsequent bootstrap modification is not claimed to provide an additional independent probabilistic guarantee for the complete final uncertainty set construction. Our theoretical results above focus on the properties of the constructed data-driven uncertainty sets and their role in the resulting RO problem. In particular, the results establish properties such as compactness, cluster construction and robust feasibility with respect to the constructed uncertainty set. The proposed bootstrapped DPMM procedure is used to construct data adaptive uncertainty sets from observed samples, but we did not claim full statistical consistency. This is therefore left as an important direction for future work.

3.4 Experiments

This section provides a series of numerical examples to test the performance of the proposed method. To implement the examples, we coded in Python with relevant libraries, such as scikit-learn for machine learning and RSOME (Robust Stochastic Optimisation Made Easy) (Chen et al., 2020) for RO. All the optimisation problems were solved with Gurobi and implemented on a university laptop with an Intel(R) Core(TM) i5-1145G7 @ 2.60GHz and 16GB RAM.

3.4.1 Motivating example – a toy problem

Here, we provide an illustrative motivating example adapted from Ning and You (2017) to test the proposed data-driven uncertainty set and the data-driven RO framework.

The motivating example is defined as follows:

$$\begin{aligned} &\min_{x} 3x_1 + 5x_2 + \max_{u \in U} \min_{y} 6y_1 + 10y_2 \\ &s.t. \\ &x_1 + x_2 \leq 70 \\ &x_1 + y_1 \geq u_1 \\ &x_2 + y_2 \geq u_2 \\ &x_1, x_2, y_1, y_2 \geq 0 \end{aligned} \tag{3.18}$$

This is a classical two-stage adaptive RO and *minmax* problem where the scope is to minimise x and y, while maximising the uncertainty u. x_1 and x_2 are the first-stage decision variables, while y_1 and y_2 are the second-stage decision variables. u_1 and u_2 are the uncertain parameters of a possible production plan that perform well across all possible realisations within the defined uncertainty set U.

Dataset and scenarios

The experimental setting for the above problem is to create two uncertain parameters, u_1 and u_2, which for example can be considered as uncertain demand in a production planning problem. We create 500 data points which are split into 3 distinct scenarios of different data structures and variability, such as symmetric, asymmetric, correlated, non-correlated patterns and no patterns. The purpose of selecting different scenario-based approaches is to capture a wide range of possible outcomes, so the data represent each time one distinct scenario. We note from equation 3.18 that potential supervised learning predictions can determine the problem's constraint boundary, which we did not expand. Our goal is to ensure the effectiveness of our approach by testing the performance of our proposed data-driven uncertainty set in terms of objective value and coverage rate. Our proposed uncertainty set was also manually scaled according to the above observed data. We primarily aim to determine if it can accurately detect and exclude outliers, as well as capture the shape of each cluster within the data and across different scenarios. We applied the same uncertainty set formulation and scaling

consistently to all clustering comparison methods. We will also compare the objective values, solutions, computational time and coverage rate. The scenarios are as follow:

- SCENARIO 1: We create 5 obvious distinct separate clusters and add some outliers - 20 data points.
- SCENARIO 2: We test the same problem and data on a more symmetric data structure with normal linear pattern.
- SCENARIO 3: We evaluate our uncertainty set on wide asymmetric data variation spread, with noise such that the data appears at irregular intervals.

Results

For SCENARIO 1, a detailed table is provided in Table 3.1 with respect to different uncertainty set construction methods, coverage rate, objective, first-stage and second-stage decisions. Different types of traditional uncertainty sets, such as U_{box} and U_{budget}, as well as other unsupervised learning methods such as k-means and DBSCAN, are constructed based on the same uncertainty data to test against our framework and approach.

Table 3.1: Comparisons of different data-driven approaches

	Box Uncertainty Set	Gamma Uncertainty Set	Proposed Method with 1 pre-defined component	DBSCAN Uncertainty Set	k-means Uncertainty Set	Proposed Method with 5 predicted exported components
Minimise Objective	**347.43**	**391.56**	**336.27**	**279.40**	**242.05**	**228.90**
First-Stage Decisions	$x_1 = 30.09$ $x_2 = 39.90$	$x_1 = 14.89$ $x_2 = 30.98$	$x_1 = 14.68$ $x_2 = 28.67$	$x_1 = 18.32$ $x_2 = 28.22$	$x_1 = 18.53$ $x_2 = 27.68$	$x_1 = 18.45$ $x_2 = 26.99$
Second-Stage Decisions	$y_1 = -1.59$ $y_2 = 0$	$y_1 = 17.64$ $y_2 = 8.61$	$y_1 = 14.93$ $y_2 = 6.91$	$y_1 = 9.11$ $y_2 = 2.86$	$y_1 = 6.97$ $y_2 = 0.61$	$y_1 = 6.19$ $y_2 = 0.13$
Coverage Rate	100%	96%	96%	93.65%	95%	95%
Computation time (s)	3	3	4 (+4 for bootstrap resampling)	4	4	4 (+20 for bootstrap resampling)

We set the *threshold* as 5%. With the proposed data-driven uncertainty set, our nonparametric unsupervised learning model correctly captures the true number of components of the data, $M = 5$. The output of the means $\{\mu\}_{k=1}^{5}$ = [13.83, 16.88], [18.39, 24.24], [10.19, 16.23], [18.16, 20.70], [12.96, 12.16], while the output of $\{\Gamma_1\}_{k=1}^{5}$ = [3.17, 3.35, 4.36, 4.66, 4.53] and for $\{\Gamma_\infty\}_{k=1}^{5}$ = [1.93, 2.47, 2.26, 2.50, 2.83], $\forall k = 1, 2, 3, 4, 5$ and $\alpha_{cov} = 100\%$ coverage rate.

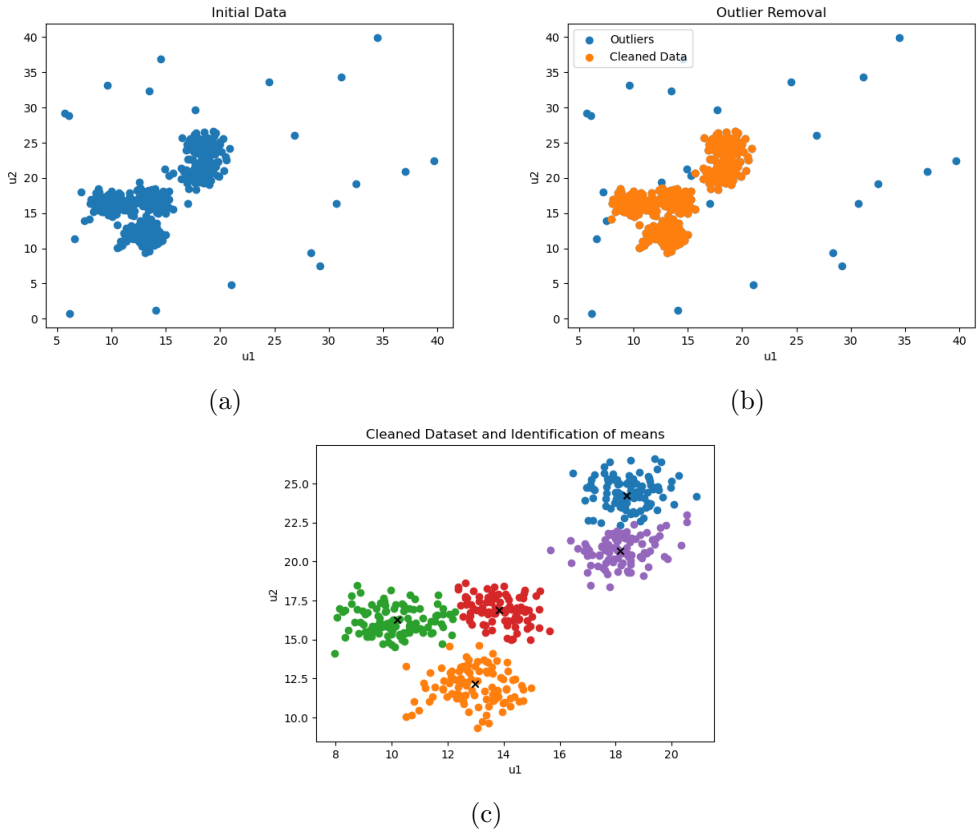


Figure 3.6: SCENARIO 1: Process for the construction of the Data-Driven Uncertainty Set. (a) initial data, (b) Outlier identification, (c) Inference in black the parameter mean of each predicted cluster

In Figure 3.6 we present the process to construct the data-driven uncertainty set, with identification of the outliers and remove them in Figure 3.6b and then our method determines the optimal number of components, with their means in Figure 3.6c for uncertain parameters u_1 and u_2.

In Figure 3.7 we experiment with the traditional uncertainty sets that are widely used in RO, Box (Figure 3.7a) and Budget/Gamma (Figure 3.7b) Uncertainty Sets, which shows excessive conservatism. In this example, the solution being on the edge makes sense because the box set pushes the solution to the limits of the variable ranges to minimise the objective function within those extreme points, from the first constraint. We also experiment our approach with different pre-defined number of components and test the impact on the constructed data-driven uncertainty set. In Figure 3.7c, after we

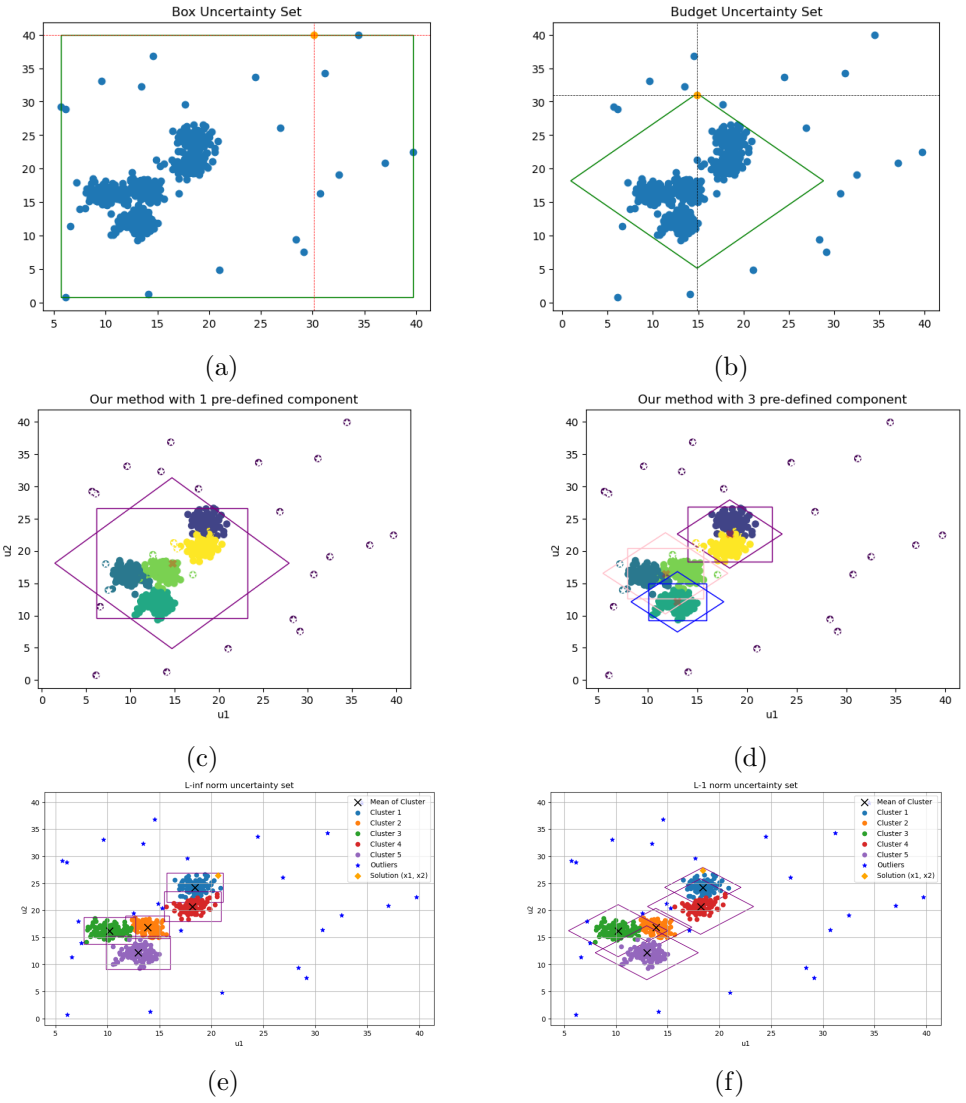


Figure 3.7: (a) and (b) are traditional Box and Budget Uncertainty Sets. (c) is our approach with 1 predefined component, (d) our approach with 3 pre-defined components. (e) is our approach with just l_∞ norm uncertainty set and (f) with just l_1 norm uncertainty set.

identify and remove the outliers noted as data points with (*), we export the uncertainty set with 1 pre-defined component, while in Figure 3.7d we export the uncertainty set with 3 pre-defined component where we can observe that it is less conservative than the previous one. In Figure 3.7e and Figure 3.7f we automatically determine the optimal number of components, where inferred number of components are 5, but we examine each norm, ℓ_∞ and ℓ_1, separately and their effects on our approach, with the same predicted number of components, means and covariances, but different Gammas. For the ℓ_∞ uncertainty set construction, we observe that the optimal first-stage decisions are positioned near the upper-right boundary of the relevant box, while the optimal recourse decisions satisfy $y_1 = y_2 = 0$. This does not imply that the second-stage variables are absent or that robustness is not introduced. In this instance, the variables are not activated because the first-stage decisions provide sufficient protection against all realisations contained within the specified ℓ_∞ uncertainty set. For instance, the bounds of the ℓ_∞ set can be covered by the first-stage decisions without violating the constraint $x_1 + x_2 \leq 70$. Furthermore, first-stage option has lower costs than recourse, since $3 < 6$ and $5 < 10$. Therefore, the objective favours covering the uncertain requirements through x_1 and x_2 rather than subsequently activating the more expensive y_1 and y_2. This zero-recourse outcome is dependent on the specific instance and case. On the other hand, positive second-stage decisions are obtained with the ℓ_1 and final combined uncertainty set constructions, which confirms our previous claim that the zero-recourse outcome is dependent on the instance. The data distribution is formed in a way that the ℓ_1 norm, $|u_1| + |u_2|$, sufficiently captures the spread and the ℓ_∞ norm, $\max(|u_1|, |u_2|)$, produces bounds that can be covered through the first-stage decisions.

Moving beyond these conventional constructions, our final bootstrapped DPMM infers 5 components for this dataset. Figure 3.8 illustrates the resulting final uncertainty set, which is constructed as the union of the intersections of the ℓ_1 and ℓ_∞ sets. In this tested scenario, the construction provides a compact representation of the observed data and produces a more central solution than the selected classical uncertainty sets. However, these findings do not imply that every possible realisation is captured. Its performance depends on the representativeness of the dataset, the clustering and the

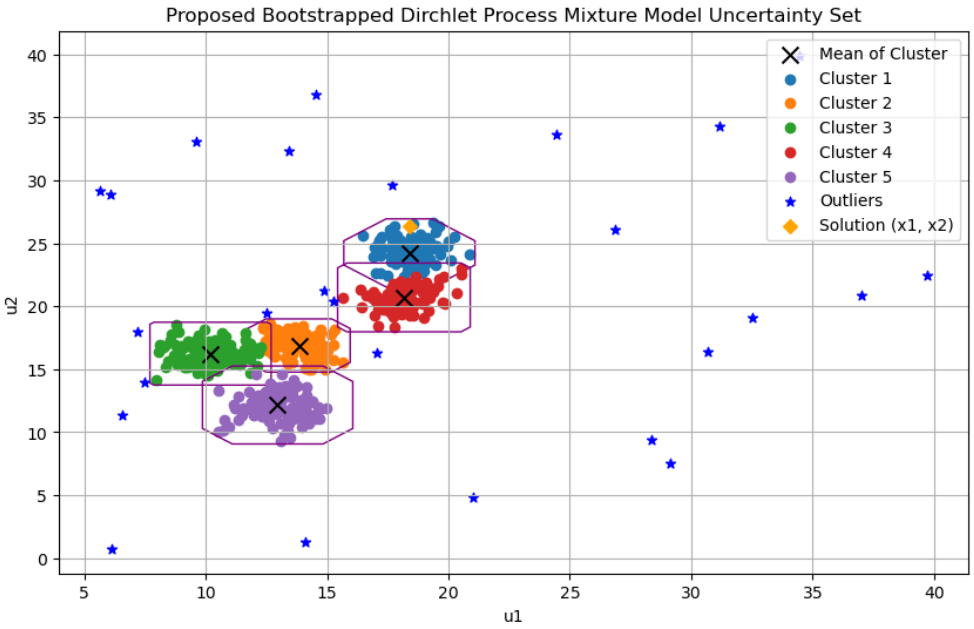


Figure 3.8: Proposed Bootstrap DPMM approach to construct the optimal number of components automatically

outlier-detection mechanism, which may exclude rare plausible adverse realisations in other scenarios. The observed compactness is a trade-off between reduced conservatism and protection against realisations outside the constructed set, alongside the additional computational cost of bootstrap resampling.

To test the power of our framework against other existing unsupervised approaches, we implemented the k-means method to infer the parameters that we integrate to the same proposed data-driven uncertainty set. Since the drawback of k-means is that we have to determine beforehand the number of components, we utilise in Figure 3.9a the "elbow method" on the same dataset without outlier, since k-means is extremely sensitive to that. We found that the optimal numbers of components can be 2, since we can split the data into two clusters (of two and three), or 5, as we correctly predicted. In Figure 3.9b we constructed the data-driven uncertainty set with k-means and we observe it is quite similar to our method. Delving deeper in Figure 3.9c, we compared the main parameters influencing the efficiency of the uncertainty set, specifically the means. We found that the centroids from the k-means algorithm differ from our DPMM centres. While k-means generally performs well when the components are distinct

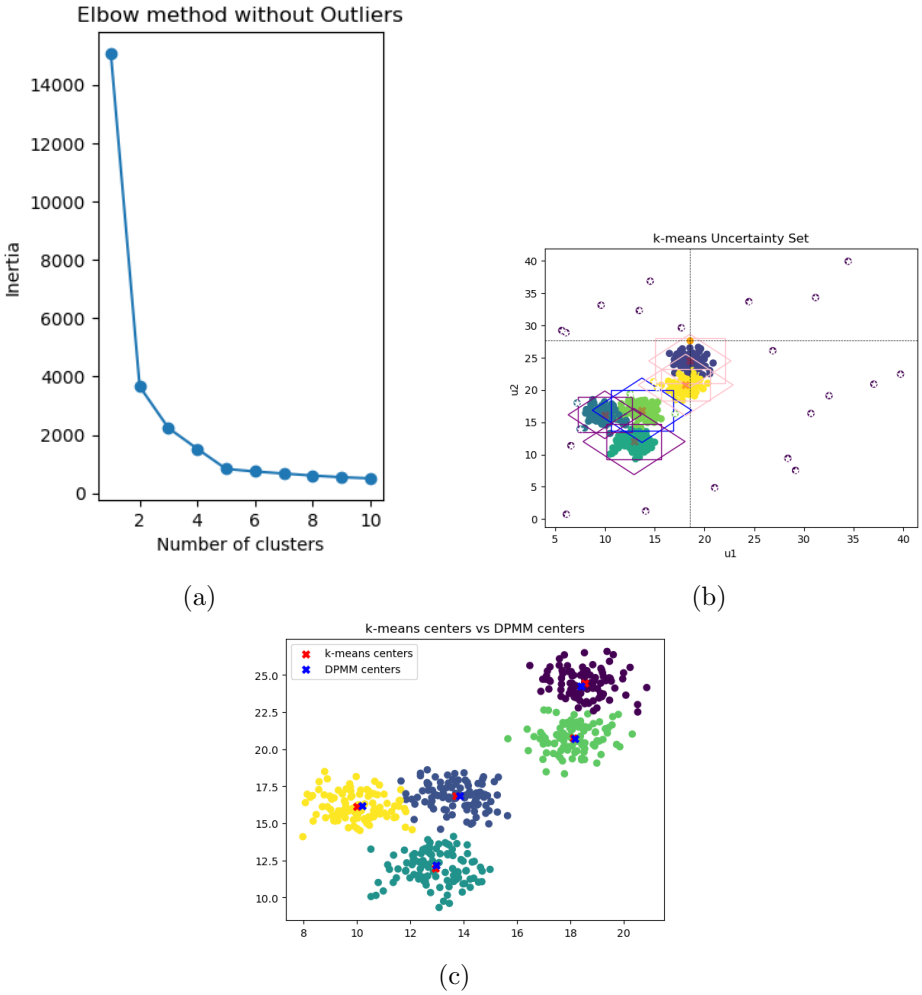


Figure 3.9: k-means algorithm, resulting uncertainty set construction and comparison

from each other, our methodology outperformed it by paying closer attention to the actual weight of the data points in each cluster and their predominant locations within each component. We anticipate k-means to fail in other scenarios where data are asymmetrical and exhibit spread variation due to its inherent drawbacks, as well as the need to find and determine beforehand the optimal number of clustering.

To further assess the effect of the clustering mechanism on the resulting uncertainty sets, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) was also evaluated in Figure 3.10 as an additional density-based baseline. DBSCAN is relevant to our comparison because, unlike k-means, it does not require the number of clusters to be fixed in advance. In addition, DBSCAN can identify low-density observations as noise

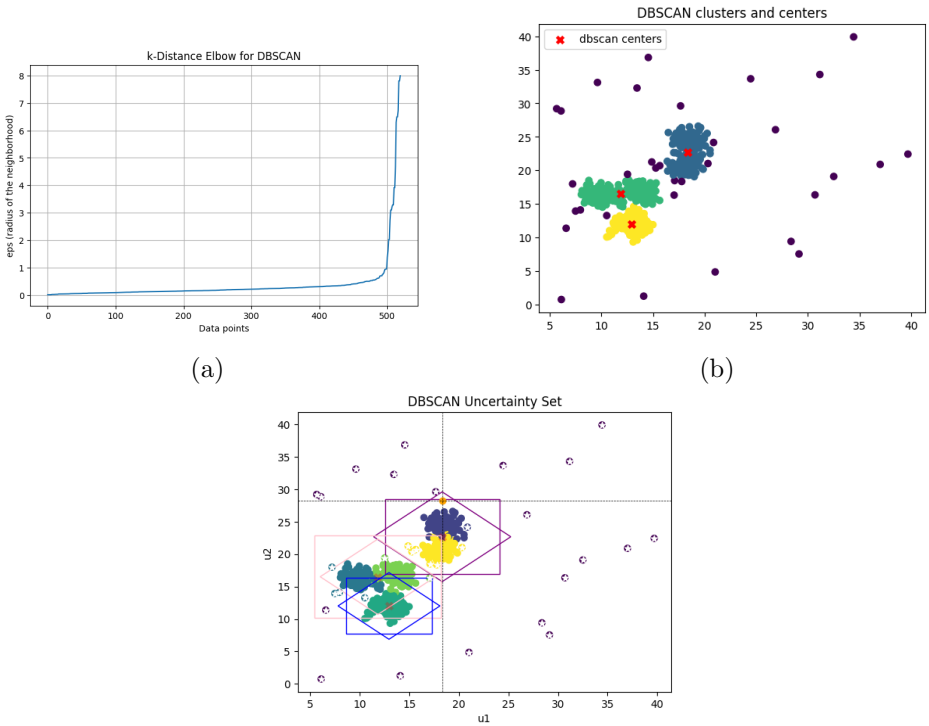


Figure 3.10: DBSCAN algorithm, resulting uncertainty set construction and comparison

which we can compare its outliers with our approach. However, the empirical results showed that DBSCAN was not well suited to the present data. The cluster structure was highly sensitive to the choice of the neighbourhood radius *eps* and *min samples*, while different choices of these parameters produced substantially different numbers of clusters and noise points. To mitigate that, we found an ideal candidate for *eps* using a classical k-distance elbow method presented in 3.10a. Thus, our choice for *eps* was 0.8 and *min samples* was 3. The outcome of DBSCAN clustering is presented in 3.10b with only 3 identified clusters, while the remaining 33 data points are flagged as outliers. Furthermore, the uncertainty set construction around those centres is presented in 3.10c with objective value of 279.40, which is competitive to k-means but also higher than our approach. Consequently, we concluded that the DBSCAN uncertainty sets led to a competitive performance but not better objective value at the desired coverage rate, especially by accounting the different clustering outcomes due to the choice of the initial

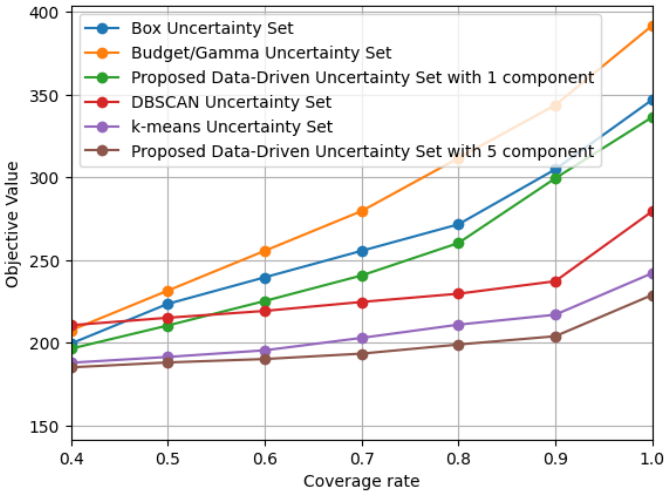


Figure 3.11: Coverage Rate Comparisons over different percentages of the same data

parameters which was challenging.

In Figure 3.11 we present the objective with different coverage rates among the traditional and proposed data-driven uncertainty sets. As shown, the proposed data-driven uncertainty set with automatically defined 5 components results in lower objective values across various coverage rates. In contrast, the other methods yield higher objective values due to the overly conservative creation of the uncertainty set and unnecessary space. The k-means method was the closest to our approach, largely because we use the same data-driven uncertainty set and adaptive scale parameters.

In the end, we compare in Figure 3.12 the bootstrapped and non-bootstrapped uncertainty set constructions. As described, bootstrap resampling estimates the sampling variability of the vital components, which is subsequently incorporated into the centres and scaling parameters of the constructed set. In this instance, the introduction of bootstrap clearly changes the uncertainty set geometry and increases the objective value from 224.74 to 228.90. This increase represents the additional conservatism introduced by accounting the estimated variability. Therefore, the result shows the observed trade-off with respect to the constructed uncertainty set and the performance of the objective value.

For SCENARIO 2, we test the same motivational example on a more symmetric data structure with a normal linear pattern. Our aim is to observe the outcome of

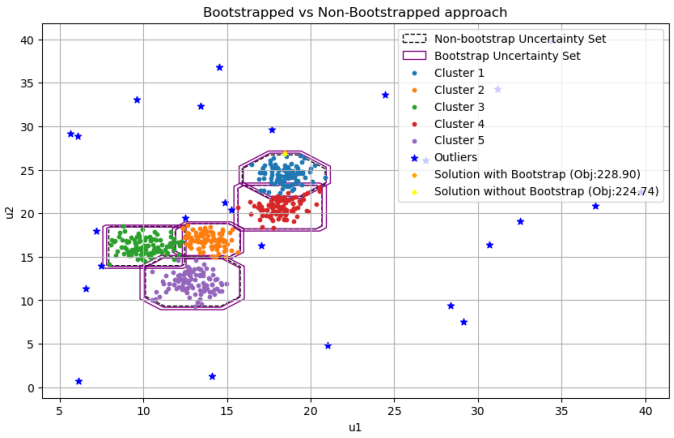


Figure 3.12: Bootstrap vs Non-Bootstrap approach: Difference between uncertainty set size, solutions and objectives

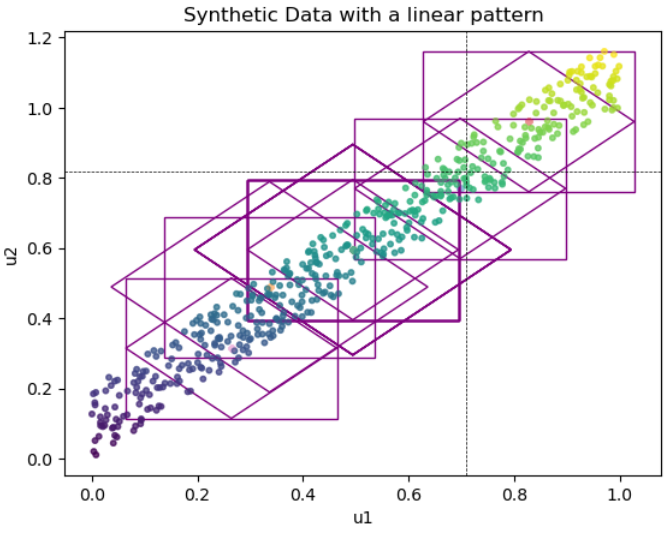
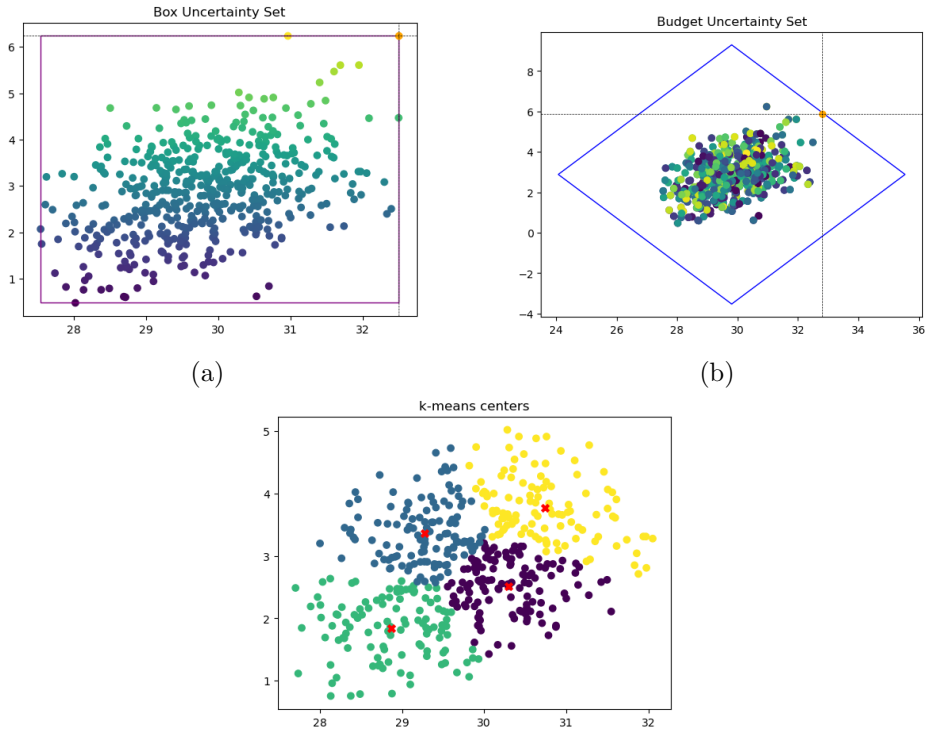


Figure 3.13: SCENARIO 2: Our framework works well on correlated dataset with linear pattern.



(a) (b) (c)

Figure 3.14: SCENARIO 3: Asymmetric dataset variation spread with noise in irregular intervals. (a) Traditional Box Uncertainty Set, (b) Traditional Budget Uncertainty Set, (c) Clustering behaviour of k-means after its own outlier detection

our proposed framework on different data structures, as shown in Figure 3.13. Once again, our method accurately captures the shape of the data, constructing a well-fitted data-driven uncertainty set that includes all necessary information. This allows us to incorporate it into the same RO problem and, for this instance, achieve the lowest objective among the conventional approaches.

Lastly, for SCENARIO 3, we will evaluate our uncertainty set on a dataset characterised by wide asymmetric variation and noticeable noise, where data appears irregularly, as shown in Figure 3.14. In Figure 3.14c, we observe different clustering behaviour from k-means model compared to our proposed approach in Figure 3.15, indicating different parameter inferences for the uncertainty set construction that influence the solutions and objectives. For this different data structure and range of uncertain parameters, our proposed method captures the observed uncertainty structure

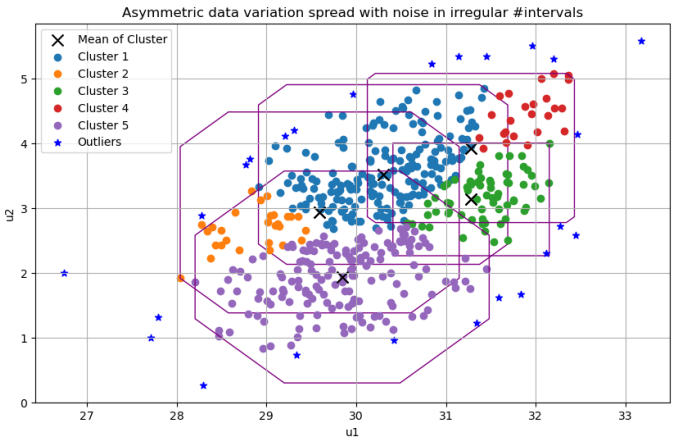


Figure 3.15: SCENARIO 3: Proposed data-driven uncertainty sets on dataset with data variation

and identifies potential outliers, while produces an objective value of 121.04, compared with 122.38 for the k-means construction. The other classical uncertainty sets produce even higher objective values. Although this is a modest difference, together with the previous scenarios, these results demonstrate consistently lower objective values than the k-means construction or other classical methods across the tested instances. However, this does not imply a general superiority.

To illustrate the behaviour of the constructed uncertainty set on held-out data, we divide the dataset into 400 training observations and 100 test observations and perform five-fold cross-validation. Figure 3.16 visually compares the held-out observations with the uncertainty set constructed from the training data. As we mentioned, the main evaluation in this chapter compares objective values across the selected uncertainty set methods and coverage rates examined previously. In the tested instance, the proposed method's testing data fall inside the trained uncertainty set, while produced lower objective values than the selected baselines at the ideal coverage rates. Further quantitative evaluation is left for future work. Together with the reported coverage rates, the train–test evaluation and cross-validation results, we provide empirical evidence that the constructed uncertainty sets capture relevant uncertainty patterns beyond the training data.

At last, for SCENARIO 1, to further test the robustness and adaptability of our

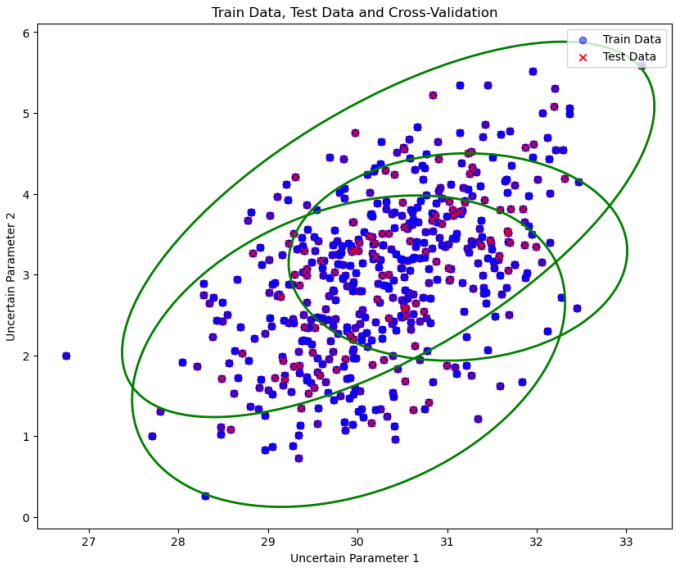


Figure 3.16: Train-test split and cross validation. Testing data fall inside the trained method and uncertainty set.

approach, we will add additional 100 data points, to see how our method's uncertainty set reacts to that, how the uncertainty modelling shifts in terms of the objective and solution in an iterative loop. Intentionally we add 2 more clear cluster to the existing dataset to evaluate if our clustering model will adapt and capture them, visualize dashed eclipses as approximation of our uncertainty set. We observe from Figure 3.17 that after 15 new data points, the model predicted and marked the majority as outliers without forming any new cluster or change solutions (orange point), but instead just adjusting the existing uncertainty set to the new modified distribution. After 35 data points, it predicted them correctly as new components and started forming 2 new clusters, changing the solution by trying to find the best approximation distribution and parameter that can express. In the end, after all the 100 data points are introduced, our method changes the solution and final uncertainty set construction representation based on the constraints, proving that it can be adaptive to new unseen data and not just solving the uncertain data without any knowledge.

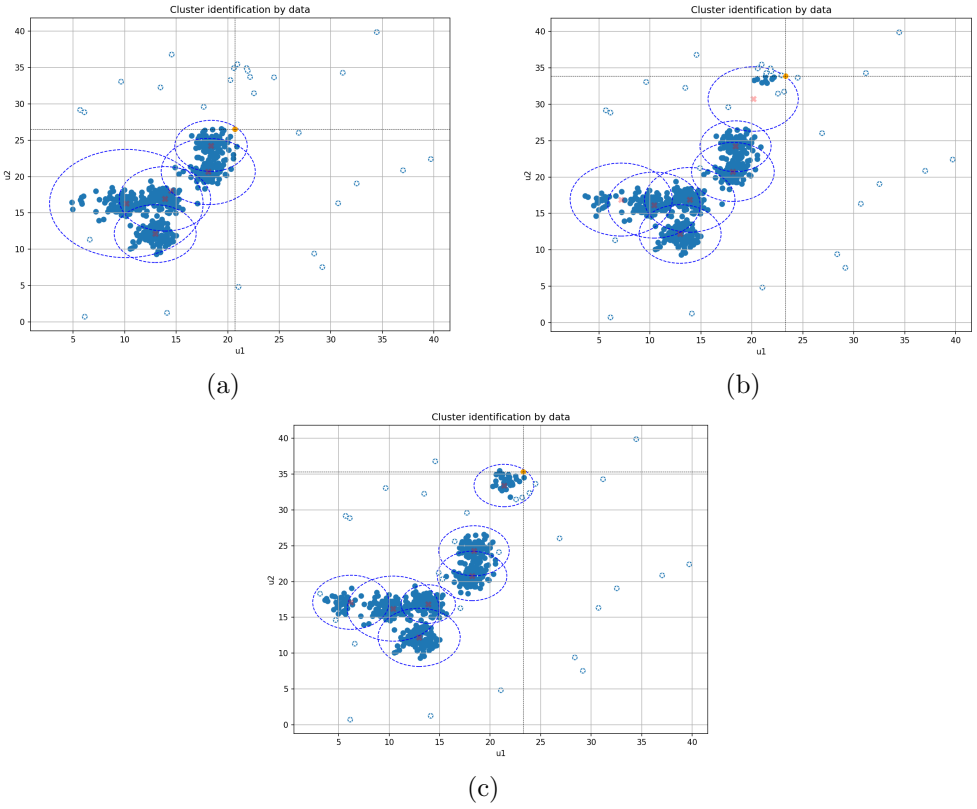


Figure 3.17: Testing on SCENARIO 1 to see how our approach behaves with additional 100 unseen out-of-sample data points over time: (a) Results after 15 new data points, (b) Results after 35 new data points, (c) Results after 100 new data points

3.4.2 Production planning example

Experimental settings

We will now consider a production planning problem in the context of a manufacturing company that produces electronic devices. The aim is to optimise a company's production decisions to meet customer demand while minimising costs, and the objective is to determine the optimal production quantities and scheduling decisions to minimise costs. To address this problem, we applied a RO technique considering uncertainties in demand, decisions, and production capacity. Our goal is to construct a compact uncertainty set by considering a range of possible scenarios and finding a production plan that performs well in all of those scenarios. See Akartunalı and Dauzère-Pérès (2022); Quezada et al. (2022) for further reading. We present a simplified formulation of the production planning problem with additional details:

1. Decision Variables:

   (a) *Production Quantities:* - x_t represents the quantity of devices produced at time period t.

   (b) *Inventory Levels:* - I_t represents the inventory level of devices at the end of time period t.

   (c) *Setup Variables:* - z_t is a binary variable that indicates whether a production setup is initiated at time period t. - v_t is a binary variable that indicates whether a production setup is completed at time period t.

2. Robust Objective Function: The objective function is to minimise the costs under this worst-case scenario. For the production and inventory costs, the robust objective function can be written as:

$$\min_{x,I} \max_{d \in \mathcal{U}} \sum_{t} \left(C_{\text{prod}} \cdot x_t + C_{\text{hold}} \cdot I_t \right)$$

   where C_{prod} is the production cost per device, and C_{hold} is the holding cost per device per time period.

3. Constraints:

   (a) *Demand Constraint:* $I_t = I_{t-1} + x_t - D_t \quad \forall t$

   (b) *Production Capacity Constraint:* $x_t \leq C_{\text{cap}} \quad \forall t$

   (c) *Setup Variables Constraint:*

   $z_t - v_t = 0 \quad \forall t, \qquad v_{t-1} - z_t \geq 0 \quad \forall t$

   (d) *Setup Time Constraint:* $\sum_t (v_t - z_t) \leq T_{\max}$

where D_t is the demand at time period t, C_{cap} is maximum production capacity and T_{max} is the maximum setup time. The traditional Box uncertainty set for the demand, as explained above, can be defined as a range that encompasses possible values. For instance, let us define the uncertain demand at time period t as D_t and define its uncertainty set as:

$$\mathcal{U}_t = \{D_t \mid D_t \in [D_t^{\min}, D_t^{\max}]\} \tag{3.19}$$

where $[D_t^{\min}, D_t^{\max}]$ represents the lower and upper bounds of the demand at time period t. But this uncertainty set as explained is highly conservative and will not give us the optimal results. As a result and for comparison purposes, we will proceed with our proposed constructed data-driven uncertainty set in Equation 3.13.

Data, analysis and results

We consider the following scenario and uncertain synthetic demand data as the experiment: We have two products A and B, such as electronic devices. In this scenario, the demand for these devices fluctuates seasonally due to changes in customer preferences and market trends. Winter and Summer months exhibit higher demands than other months, over a period of 365 days. The demand D_t for each time period t is uncertain and vary. The minimum demand for a period t is denoted as $D_t^{\min}$ and the maximum $D_t^{\max}$. We set the production cost per device to $C_{\text{prod}} = 100$ and the holding cost per device per time period $C_{\text{hold}} = 10$. The production capacity is determined $C_{\text{cap}} = 1000$ per time period t and the maximum setup time predicted is $T_{\max} = 3$. We have also experimented with different numerical values for these. For each time period t, d_1 represents the uncertain demand for electronic device A and d_2 represents the uncertain

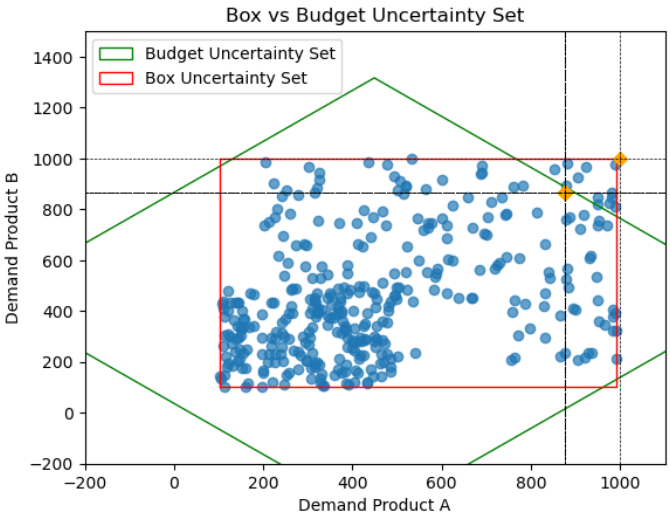


Figure 3.18: Traditional Uncertainty Sets for Production Planning experiment. Box vs Budget Uncertainty Sets

demand for electronic device B. In this experiment, we will have 12 solutions for 12 days for each time period $t = 1, 2, ..., 12$, instead of 1 like a usual RO approach.

Initially, we addressed the experiment using conservative approaches with a box uncertainty set and Budget uncertainty set. In Figure 3.18, the visualisation of the data over the 12 time periods, derived using the classical uncertainty sets, are shown. The overall 12 solutions (orange points) for both Product A and Product B using the Box and Budget Uncertainty set are shown there. Our framework chose to produce at the same maximum demand for each time period for each product, highlighting an extremely conservative approach. Budget approach performed slightly better than Box, but it was still highly conservative, disregarding seasonal fluctuations in demand and resulting in extremely conservative approach.

Finally, the results of our approach are presented in Figure 3.19. Our ML algorithm initially captured the demand structure by approximating the data distribution variability and proposing the optimal number of components, the mean and covariance of each cluster, along with Gammas of l_1 and l_∞ norms for each component. This approach produced a less conservative uncertainty set, avoiding unnecessary space, identifying outlier (marked as "*") and infeasible scenarios/solutions. For this experiment, the proposed uncertainty sets were manually constructed from scratch and adjusted

Table 3.2: Results and Comparison with existing approaches

	Product A	Product B
Demand	d_A = [876, 638, 666, 522, 774, 314, 514, 424, 212, 814, 450, 388]	d_B = [234, 450, 452, 486, 608, 866, 860, 500, 232, 786, 258, 580]
Minimum Demand	$(d_A)_{min} = 212$	$(d_B)_{min} = 232$
Maximum Demand	$(d_A)_{max} = 876$	$(d_B)_{max} = 866$
Minimum/Maximum Demand estimated after final period	[2544, 10512]	[2784, 10392]
Objective with Box Uncertainty Set	**1,567,879**	**1,532,380**
Initial Solution with Box Uncertainty Set	[752, 1000, 876, 876, 876, 876, 876, 876, 876, 876, 876, 876]	[732, 1000, 866, 866, 866, 866, 866, 866, 866, 866, 866, 866]
Inventory Levels with Box Uncertainty Set	[752, 1752, 2628, 3504, 4380, 5256, 6132, 7008, 7884, 8760, 9636, 10512]	[732, 1732, 2598, 3464, 4330, 5196, 6062, 6928, 7794, 8660, 9526, 10392]
Objective with Gamma/Budget Uncertainty Set	**1,219,960**	**1,207,860**
Initial Solution with Gamma Uncertainty Set	[1252, 1000, 876, 876, 876, 876, 876, 876, 876, 876, 876, 876]	[1232, 1000, 866, 866, 866, 866, 866, 866, 866, 866, 866, 866]
Inventory Levels with Gamma Uncertainty Set	[1252, 2252, 3128, 4004, 4880, 5756, 6632, 7508, 8384, 9260, 10136, 11012]	[1232, 2232, 3098, 3964, 4830, 5696, 6562, 7428, 8294, 9160, 10026, 10892]
Objective with Proposed Data-Driven Uncertainty Set	**705,537.84**	**655,962.93**
Initial Solution with Proposed Data-Driven Uncertainty Set	[0, 678.78, 860.61, 392.53, 409.75, 230.54, 447.27, 473.49, 416.86, 784.74, 807.53, 750.36]	[0, 523.86, 860.61, 392.53, 409.75, 230.54, 447.27, 473.49, 416.86, 784.74, 807.53, 750.36]
Inventory Levels with Proposed Data-Driven Uncertainty Set	[0, 678.78, 1539.39, 1931.93, 2341.68, 2572.22, 3019.50, 3492.99, 3909.85, 4694.60, 5502.14, 6252.50]	[0, 523.86, 1384.47, 1777.01, 2186.76, 2417.30, 2864.57, 3338.07, 3754.93, 4539.68, 5347.22, 6097.58]

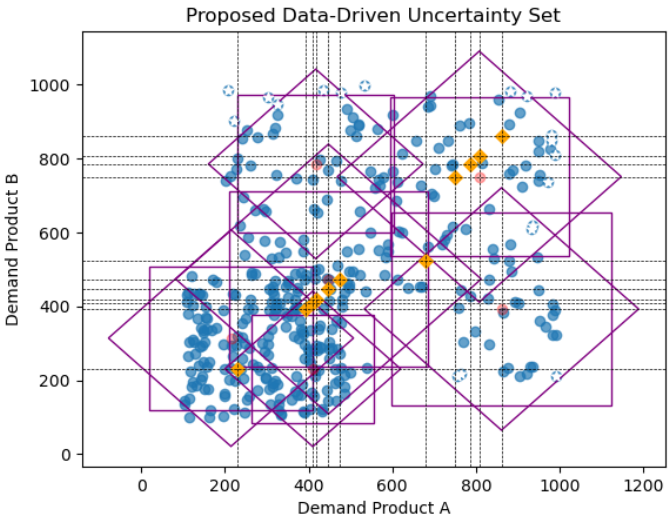


Figure 3.19: Proposed data-driven approach for experiment 2 over different time periods (t=12)

accordingly based on the data. As such, our methodology identified the best predicted demand for each product, accounting for seasonal fluctuations and the original demand's possible interval levels.

When implemented, the objectives, first-stage decisions and inventory levels are presented in Table 3.2 for Product A and Product B. The solutions remained within the bounds of the constructed data-driven uncertainty sets, where bootstrap produce more robustness, ensuring no solutions fell outside the range of actual production and data points. The framework adjusted and adapted for subsequent 12 time periods, providing the best solutions for both electronic products. In addition, the objectives for the demand of Product A and Product B were significantly lower compared to the previous highly conservative methods. Lastly, the inventory level is lower due to the varying production in each period (t), which accounts for demand fluctuations. This approach is less conservative but remains robust across the different scenarios.

3.4.3 Robust vehicle pre-allocation example

In this example, we solve a modified vehicle pre-allocation problem, initially introduced by Hao et al. (2020). Given I supply nodes and J demand nodes in an urban area,

after random demand $d_j, j \in [J]$ is realised, the operator allocates x_{ij} vehicles from supply node $i \in [I]$ to demand node $j \in [J]$ at a unit cost c_{ij}. The detailed parameters are listed as below:

- Number of supply nodes I=1
- Number of demand nodes J=10
- Revenue coefficients
  r =(4.50,4.41,3.61,4.49,4.38, 4.58,4.53,4.64,4.58,4.32)
- Cost coefficients c_j=3 , where j=1,2,. . . ,J
- Maximum supply of vehicles q_i=400, where i=1,. . . ,I

The vehicle pre-allocation was solved by the robust and sample RO (Chen et al., 2020), where the decision under demand uncertainty was formulated by solving the RO problem below:

$$
\begin{aligned}
&\min_{x,y} \max_{u \in U} \left(\sum_{i=1}^{I} \sum_{j=1}^{J} (c_{ij} - r_j) x_{ij} + \sum_{j=1}^{J} r_j y_j \right) \\
&s.t. \\
&y_j \geq \sum_i x_{ij} - u_j \qquad \forall u \in U, \forall j \in [J], \forall i \in [I] \\
&\sum_j x_{ij} \leq q_i \\
&x_{ij}, y_j \geq 0
\end{aligned}
\tag{3.20}
$$

where the wait-and-see decision y represents the bookkeeping revenue. It was initially solved with a conservative box uncertainty set where the upper and lower bounds are identified using a sample demand dataset ($taxi - rain.csv$). The dataset includes the demands of 10 regions associated each revenue coefficient and 77 day time demands.

In this experiment, the proposed uncertainty set was manually constructed from scratch and adjusted accordingly based on the data. We compare with two alternative uncertainty set construction approaches (Chen et al., 2020; Hao et al., 2020): the sample average approximation approach with the conservative Box uncertainty set, defined

Table 3.3: Robust Vehicle pre-allocation Results

	Box Uncertainty Set	ϵ-neighborhood Uncertainty Set	Proposed data-driven uncertainty set construction
Objective	-62.59	-103.65	-108.55
Pre-Allocation Decisions	[0, 0, 0, 0, 0, 39.61, 0, 0, 0, 0]	[0.34, 0.35, 0, 4.28, 0, 69.45, 2.45, 4.57, 5.22, 2.48]	[0, 0.11, 0, 1.55 , 0, 75.86, 0.52, 2.76, 1.74, 0]
Coverage rate	100%	100%	100%
Computational Time	10 sec	13 sec	31 sec

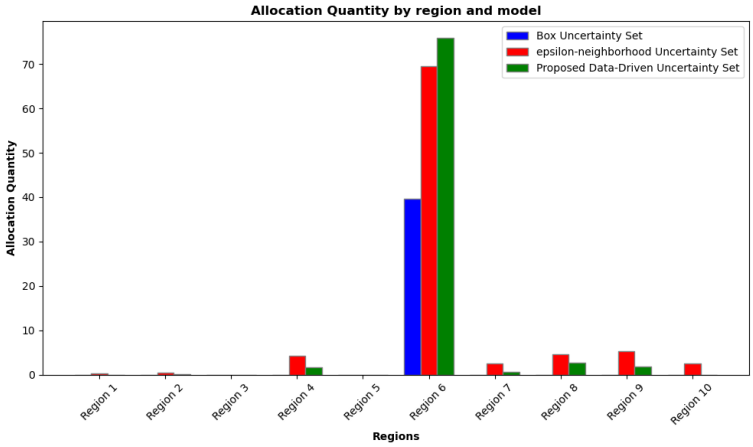
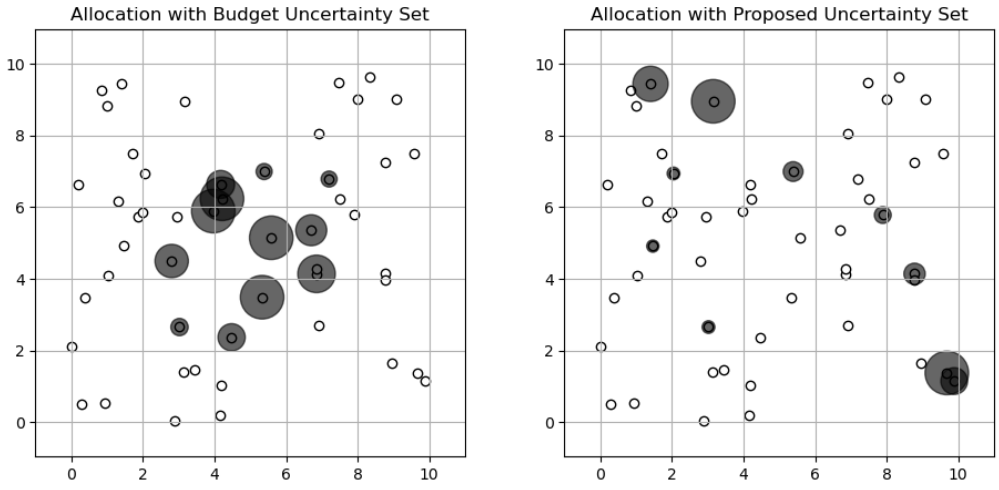


Figure 3.20: Experiment 3: Allocation Quantity by region and comparison between Box vs epsilon-neighborhood vs proposed data-driven uncertainty sets



(a) (b)

Figure 3.21: Experiment 4: Lot sizing allocation. Difference allocation between (a) Budget uncertainty set and (b) proposed bootstrapped DPMM uncertainty set.

as $U_{box} = (d|\ \ d_{min} \leq d \leq d_{max})$, and the two-stage sample robust model approach with the ϵ-neighborhood uncertainty set, which is defined $U_\epsilon = \left(d|\ \ d_{min} \leq d \leq d_{max},\ \ \|d - \hat{d}_s\| \leq \epsilon\right)$ for a collection $\left(\hat{d}_1, \hat{d}_2, \cdots, \hat{d}_S\right)$, given in Bertsimas et al. (2022). In Table 3.3 we present the results and the existing approaches in comparison with our algorithm and data-driven uncertainty set construction. We also demonstrate in Figure 3.20 the difference in quantity allocation for each model for each region. The results for this specific experiment indicate that our proposed method performs better than the conservative sample average approximation approach and slightly better than the sample robust approach only in terms of objective value and for similar coverage rates. However, although our approach exhibits less conservative and more precise allocation, it is computationally more expensive especially in such small dataset. This is due to the bootstrap resampling that increases the computational burden.

3.4.4 Adaptive robust lot-sizing example

In this experimental setting, we examine a modified lot-sizing problem, initially proposed by Bertsimas and de Ruiter (2016) and revisited by Chen et al. (2020). Within a network comprising N stores, the allocation of stock, represented by x_i for each store

i, is established prior to the realisation of the actual demand at each location. The demand, d, is subject to uncertainty and presumed to belong to a budget uncertainty set.

Following the observation of the demand realisation, stock y_{ij}, the transportation from store i to j, aims to meet all demand with an associated cost t_{ij}. The goal is to minimise the total cost. The proposed uncertainty set and formulation of this example is given as below:

$$
\begin{aligned}
& U = (d : 0 \leq d \leq d_{max}, \;\; d \leq \Gamma) \\
& \\
& \min_{x,y} \max_{d} \sum_{i=1}^{N} c_i x_i + \sum_{i=1}^{N} \sum_{j=1}^{N} t_{ij} y_{ij} \\
& s.t. \\
& d_i \leq \sum_{j=1}^{N} y_{ji} - \sum_{j=1}^{N} y_{ij} + x_i \quad d \in U \\
& 0 \leq x_i \leq K_i \quad i = 1, ..., N
\end{aligned}
\tag{3.21}
$$

For comparisons we set $N = 50$ locations and storage cost $c_i = 20$. t_{ij} is given from the Euclidean distance between the two stores. d_{max} is the maximum demand and K_i is the stock capacity both set to 20. The target is to identify the appropriate stock allocation at each location. The results are presented in Table 3.4 and the different visualisation in the vehicle allocations for each store in Figure 3.21. For this experiment, the proposed uncertainty set was manually constructed from scratch and adjusted accordingly based on the data.

The experimental results demonstrate improved objective value of our proposed data-driven uncertainty set in optimising stock allocation within the modified lot-sizing problem. Despite the increase in computational time, our method consistently outperforms the traditional budget uncertainty set in terms of objective cost, achieving a lower total cost of 3633.19 compared to 3647.01. Both methods kept 100% coverage rate, while the complete decision vectors are not included because the model contains $N = 50$ first-stage stock allocation variables and way more second-stage variables.

Table 3.4: Adaptive Robust Lot-Sizing Results

	Budget Uncertainty Set	Proposed data-driven uncertainty set construction
Objective	3647.01	3633.19
Coverage rate	100%	100%
Computational Time	27 sec	58 sec

Both methods use the same two-stage recourse formulation and differ only in their uncertainty set construction. The findings of this study clearly show that even though we slightly outperform and cannot fully leverage the unsupervised learning capabilities with only a limited amount of available data, the integration of Bootstrapped DPMM in constructing uncertainty sets enhances the decision-making process when faced with uncertain environments.

3.5 Concluding remarks

In this chapter, we provided a novel methodology to construct compact uncertainty sets for RO, specifically by employing a data-driven approach based on the bootstrapped Dirichlet Process Mixture Model. We first introduced the static RO formulation and then focused primarily on finite stage decision-making, with a limited number of decision stages, specifically two-stage RO, where uncertainty was used to be represented by fixed uncertainty sets. Unlike other classical uncertainty set constructions, our method provides a far greater flexibility and significantly reduces unnecessary conservatism. We concluded that the union of l_1 and l_∞ norms, along with the powerful DPMM and the innovative introduction of bootstrapped resampling, led to promising results in the field. Particularly, computational experiments demonstrated that RO solutions obtained from our approach consistently outperformed traditional uncertainty models. This chapter bridges the gap between predictive machine learning and prescriptive RO, mainly by showing the practical advantages of employing advanced data-driven tech-

niques within the uncertainty set construction. In the next chapter, we provide an extension and, in particular, an implementation of a fully sequential decision-making framework. Furthermore, we present how the introduction of statistical, data-driven and machine learning methods can improve the uncertainty set construction within that context as well.

Chapter 4

Compact Bayesian Ambiguity Sets for Robust Markov Decision Processes

In the previous chapter, we introduced data-driven uncertainty set construction for RO, first through a static formulation and then primarily through two-stage RO, where we demonstrated improved performance in several and diverse instances, mainly in terms of objective value. This was accomplished by employing machine learning models in the vital uncertainty set construction. However, both formulations primarily address the RO setting. We therefore turn to a fully sequential domain and show how a similar data-driven philosophy can be extended by constructing ambiguity sets for RMDPs within the RRL framework.

4.1 Introduction

In recent years, RL has demonstrated a remarkable success in various applications (Sutton, 2018). However, the practical deployment of RL agents in real-world environments is limited by their sensitivity to uncertainties. Traditional RL methods are effective in controlled settings, but they fail miserably to maintain consistent performance under model misspecifications (Moos et al., 2022), which implies that if the

underlying distribution changes after the training process, the agent is trained to the wrong distribution and thus fails to select the correct actions (Decker and Sester, 2024). Furthermore, most of the times, we assume that transition probabilities are fixed or known exactly. However, this assumption leads to errors due to value function approximations, modelling errors, and mainly due to limited or noisy data (Petrik and Russel, 2019; Petrik and Subramanian, 2016). The challenge of properly handling the true nature of uncertainty has generated a strong interest in developing techniques that can ensure high-quality performance under such unpredictable conditions. Consequently, RMDPs provide a foundational framework for enhancing the robustness of these agents (Petrik and Russel, 2019; Tirinzoni et al., 2018; Wiesemann et al., 2013) and computing policies that offer protection against transition uncertainty. Essentially, we model transitions as elements within an uncertainty set, commonly called ambiguity set. This ambiguity set allows for constructing policies that are optimised for worst-case scenarios, specifically to construct an optimal robust policy that achieves the best performance under the worst possible model within this ambiguity set (Panaganti et al., 2022). However, existing ambiguity set construction methods produce overly conservative solutions by relying on fixed datasets with limited coverage of critical state–action pairs. An essential aspect is to find a balance between robustness and performance, where large ambiguity sets can lead to overly conservative policies and small sets result in sensitivity to model estimation errors (Derman et al., 2020; Petrik and Russel, 2019).

While promising progress has been made in improving the robustness of policies, several important challenges remain. Initially, the traditional approaches to construct ambiguity sets solely rely on concentration inequalities and confidence regions (Auer et al., 2008; Ghavamzadeh et al., 2016; Moos et al., 2022; Petrik and Russel, 2019; Thomas et al., 2015; Weissman et al., 2003). Furthermore, they often assume that the dataset is fixed and already collected in a batch setting, mainly for time-consuming reasons (Petrik and Russel, 2019), or just use a simple exploration, if any at all. As a result, this leads to conservative solutions, along with the insufficient adaptation to high-quality observed data for the main components of the ambiguity set: size (radius) and location (centre). Specifically, the location of $\mathcal{P}_{s,a}$ refers to its centre, which is the

nominal transition vector $\bar{p}_{s,a}$, while its size refers to the ℓ_1-radius $\psi_{s,a}$. An inaccurate centre or an unnecessarily large radius may result in more pessimistic robust values and policies. Both key elements will be estimated in the following subsection.

Furthermore, conservativeness is defined through set inclusion. For ambiguity sets satisfying $\mathcal{P}^{(1)} \subseteq \mathcal{P}^{(2)}$, which means that every transition model in the first set is also contained in the second, we observe that the second set is weakly more conservative because, for any fixed policy π,

$$\min_{P \in \mathcal{P}^{(2)}} V_P^\pi(s) \leq \min_{P \in \mathcal{P}^{(1)}} V_P^\pi(s)$$

since $\mathcal{P}^{(2)}$ contains more possible models, thus its worst-case return can only be equal to or lower than $\mathcal{P}^{(1)}$.

In general, the fundamental challenge in robust decision-making under uncertainty is to properly construct ambiguity sets that are both safe and not overly conservative, especially when data is limited. It is known that an ineffective ambiguity set construction has major drawbacks, such as containing implausible state transitions (Hwang and Hong, 2024). Moreover, that ambiguity set should have the correct structure, for both size and location, be tractable and accurately capture the inherent uncertainties. In Deep Reinforcement Learning (DRL) (Zhou et al., 2020), accurately representing uncertainty becomes particularly challenging and computationally intractable, as deep networks involve thousands of parameters. Consequently, it is extremely difficult to capture the full uncertainty over Q-values of deep networks.

On the other hand, the true nature of a transition probability is also unknown, which is precisely why the ambiguity set is needed, and uncertain, often occurring from limited data. Limited data leads to inaccurate estimates due to the large uncertainty involved in (s,a) pairs that are rarely or never visited, thus limited coverage. This is exactly where the upcoming robust planning is more sensitive. As a result, when we use these incorrect transition probabilities to compute a policy, we result in brittle policies that fail miserably when deployed in real-world domains (Petrik and Russel, 2019). Therefore, the robustness in RL is vital and thus the goal of our work is to compute policies that are robust, but not being too conservative and actually account

the uncertainties.

In this chapter, we propose an RMDP framework, an applied methodological extension, that involves robust exploration to generate a robust set of trajectories and then construct a compact Bayesian ambiguity set using the previously collected transition data. Building on established Bootstrapped DQN and Bayesian ambiguity set methods, this chapter presents an incremental methodological integration in which Bootstrapped DQN supports the exploratory transition data collection, while Dirichlet posterior samples are used to construct state-action ℓ_1 ambiguity sets for robust planning. We adopt the RMDP framework because our primary source of uncertainty are transition and model uncertainty, arising from limited or noisy data, and potential model misspecification. As a result, RMDPs provide an explicit way to represent this uncertainty through an ambiguity set over transitions, while safety is required under model misspecification. Our novelty lies in this integration of an innovative smart exploration before the vital ambiguity set construction in a Bayesian manner.

Several works rely on pre-collected data (Petrik and Russel, 2019; Petrik and Subramanian, 2016), which is convenient but also quite challenging. In our work, we choose how to collect the transition data in an online deep exploration within a simulator. Particularly, we utilise a robust exploration technique using a modified bootstrapped deep Q-network, essentially an ensemble Q-learning algorithm, to generate robust transitions, which are stored in the transition dataset (Osband et al., 2016). These data-driven collected transitions influence the location and position of the ambiguity set. Furthermore, its counts are used to update the prior and produce a posterior distribution that directly reflects the transition probabilities of the environment. Finally, the posterior of the bootstrapped transition probabilities is integrated to tackle the critical research gap: constructing a compact ambiguity set that reduce conservativeness. Consequently, we ensure that the learned policy is robust to transition uncertainty while outperforming other classical methods. The research problem of tight ambiguity set construction has also limited practical use in production planning environments.

Generally, uncertainty-aware exploration is well established in RL. In this chapter, however, the uncertainty represented by the Bootstrapped DQN heads is used only

to guide the exploratory transition-data collection. To the best of our knowledge, the integration of efficient smart exploration before the ambiguity set construction is understudied. The incremental contribution therefore lies in this specific sequential integration, rather than in combining exploration and uncertainty estimation in general.

4.2 Problem Statement

Motivated by the limitations of existing ambiguity set constructions, the objective of this chapter is to develop a data-driven RMDP framework that constructs compact and tractable ambiguity sets to improve the robustness–performance trade-off. This leads to two main research questions: **(RQ.1.)** *how can uncertainty information from Bootstrapped DQN variant be used in the Bayesian ambiguity set construction?* and **(RQ.2.)** *does the resulting data-driven ambiguity set construction improve the worst-case return of RMDP?* These questions define the objectives of the chapter.

We initially formulate the decision-making problem as a MDP based on a fixed and already collected dataset. A MDP is generally characterised by the tuple (S, A, r, P, γ), where S is the state space, A the action space, $r : S \times A \to \mathbb{R}$ is the reward function, and $\gamma \in (0, 1)$ is the discount factor. $P_{s,a}(s')$ is the function that defines the probability of transitioning to state s' given that action a is executed in state s. Here, we will focus on a finite MDP, that implies both $|S|$ and $|A|$ are finite sets. Additionally, we assume that the reward function is known, is bounded such that $r(s, a) \in [0, 1]$ for every state-action pair $(s, a) \in S \times A$, and is fixed.

A deterministic policy π defines an action for each state. The value function of a policy π for an MDP with a given model P, evaluated at a state s, is defined as:

$$V_P^\pi(s) = \mathbb{E}_{\pi,P}\left[\sum_{i=0}^{\infty} \gamma^i r(s_i, a_i) \mid s_0 = s\right] \tag{4.1}$$

where a_i is the the sequence of actions that follows the policy π. The next state s_{i+1} is determined according to the transition probability $P_{s_i,a_i}(\cdot)$. In standard MDP with a known fixed P, the optimal value function V_P^* and the corresponding optimal deterministic policy π_P^* under model P are then given by:

$$V_P^* = \max_\pi V_P^\pi \qquad \pi_P^* = \arg\max_\pi V_P^\pi \tag{4.2}$$

The Q-value function is:

$$Q_P^\pi(s,a) = \mathbb{E}_{\pi,P}\left[\sum_{i=0}^{\infty} \gamma^i r(s_i,a_i) \mid s_0 = s, a_0 = a\right] \tag{4.3}$$

The return policy $\rho(\pi, P) = P_0^\top V_P^\pi$, where P is the transition probability that we need to calculate and P_0 is the initial distribution. In general, the target is to calculate such policy $\pi : \mathcal{S} \to \mathcal{A}$ which maximises the return $\rho(\pi, P^\star)$, where $P^\star$ the true transition probability. However, this $P^\star$ is unknown and infeasible to compute directly in real-world settings, which is precisely why the ambiguity set is valuable. Particularly, the transition dynamics and transition probabilities are unknown and must be inferred from data, in contrast to simulated environments where the transition model is predefined. As a result, data-driven approaches are essential to approximate optimal decision-making under uncertainty.

In contrast to the standard MDP, which focuses on a single model defined by a transition probability function, in RMDP we consider diverse possible models, forming what is known as the uncertainty set–or ambiguity set in RRL–denoted by $\mathcal{P}$. This ambiguity set holds all the possible models that we may encounter. RMDPs are NP-hard to solve, but they are tractable for different ambiguity sets (Lobo et al., 2020). The simplest such type is the sa-rectangular ambiguity set which means that the transition probabilities corresponding to each state-action pair are independent. From Petrik and Russel (2019), we assume that this set satisfies the rectangularity condition, in order to simplify the analysis.

Therefore, the RMDP described by the updated tuple $(S, A, r, \mathcal{P}, \gamma)$. Specifically, the uncertainty set $\mathcal{P}$ is defined as multiple $\mathcal{P}_{s,a}$ sets for each state-action pair:

$$\mathcal{P} = \prod_{(s,a)\in\mathcal{S}\times\mathcal{A}} \mathcal{P}_{s,a} \tag{4.4}$$

where $\mathcal{P}_{s,a}$ has transition probabilities $P_{s,a}$ that are close to the nominal model $\bar{\mathcal{P}} = (\bar{P}_{s,a})$, within a specified distance $D(P_{s,a}, \bar{P}_{s,a}) \leq c_r$. Here, $\bar{P}_{s,a}$ is a single reference

estimate of the transition kernel used as the centre of the ambiguity set and it represents the location, while $c_r > 0$ determines the size and robustness level. The function $D(\cdot, \cdot)$ is a metric that measures the distance between two transition probabilities, and it essentially implies how much the actual model can deviate from the nominal model.

The transition sample dataset $\mathcal{D}$ is provided:

$$\mathcal{D} \subseteq \{(s, a, s') : \quad s, s' \in \mathcal{S}, a \in \mathcal{A}\} \tag{4.5}$$

In general, $\mathcal{D}$ can be limited, pre-collected or fixed dataset of transitions obtained from prior historical data, which is not informative enough. Because the challenge and target is to construct compact ambiguity set and we are in simulated environments, we will also make the following assumption about $\mathcal{D}$ from Petrik and Russel (2019) and Lobo et al. (2020), that the next state $s' \in \mathcal{S}$ in $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ is distributed according to the true transition probability $s' \sim P^\star(s, a, \cdot)$. However, during learning, we are only estimating the nominal transitions from observed counts. Even though in real world settings $P^\star$ is unknown, in our simulated benchmarks a generative environment samples the next state from $P^\star$, which is a fixed but unknown transition. It is used only to generate trajectories and for benchmarking reasons because the efficient computation of (s, a, s') in $\mathcal{D}$ is vital.

The objective of solving an RMDP is to define an optimal robust policy, which maximises the expected value even under the worst-case scenario within the ambiguity set. Specifically, the robust value function V^π for a given policy π, and the optimal robust value function V^*, are defined as follows:

$$V^\pi = \inf_{P \in \mathcal{P}} V_P^\pi \qquad V^* = \sup_\pi \inf_{P \in \mathcal{P}} V_P^\pi \tag{4.6}$$

For an optimal robust policy π^*, the robust value function equals the optimal robust value function, $V^{\pi^*} = V^*$.

The robust Bellman equation is:

$$V(s) = \max_{a \in A} \min_{p \in \mathcal{P}} \left\{ r(s, a) + \gamma \sum_{s'} p(s'|s, a) V(s') \right\} \tag{4.7}$$

where π represents the deterministic policy and $\mathcal{P}$ is the constructed data-driven ambiguity set. Our focus is restricted to deterministic policies.

As discussed, the aim of our online exploration is to challenge the issue of limited or fixed dataset, and thus gather a diverse and robust dataset $\mathcal{D}$ for equation 4.5. The empirical transition probabilities $\bar{P}$, which is the nominal model obtained from $\mathcal{D}$, approximates the true transition dynamics P^* as closely as possible. Furthermore, the uncertainty in our setting is sourced from the limited sample counts, not because P^* is unknown, but due to the fact that we only have a small number of samples to estimate the ground truth. We are uncertain because we have only seen a small number of transitions sampled from P^*, thus we have limited data to construct our ambiguity set. As mentioned, this includes large uncertainty because some (s,a) pairs have limited attention and are not often visited. That is the reason why we need to generate more data, such as robust trajectories and observations, which will lead to our estimated $\bar{P}$, the centre of our ambiguity set.

Finally, in order to properly find a balance between performance and robustness, we need such a statistical method, essentially a parameter that controls the tightness, which is the radius $\psi_{s,a}$, of the ambiguity set for each state-action pair. Although controlling this radius is not equivalent to the performance-robustness trade-off, the radius influences this trade-off: an unnecessarily large radius may produce overly pessimistic robust values, whereas an excessively small radius may provide insufficient coverage of the true transition dynamics $P^\star$. Therefore, our method seeks to reduce unnecessary conservativeness while keeping the posterior coverage level, rather than applying a common fixed or conservative radius. Here, robustness refers to optimisation against the worst-case transition model within the ambiguity set. To the best of our knowledge, prior works have not combined this simplified Bootstrapped DQN exploration strategy with RMDP training in this statistical manner. Solving the resulting RMDP produces the learned robust policy and its robust value function.

4.3 Methodology

This section explains our framework and approach which involve 3 vital steps. The first step is when the agent interacts with the environment using the simplified Bootstrapped DQN to collect informative transition data. At each step, the agent selects an action a_i, after which the environment generates the next state according to $s_{i+1} \sim P^\star(\cdot \mid s_i, a_i)$. The true transition distribution $P^\star$ is unknown to the agent, while the agent observes only the transitions stored in the dataset $\mathcal{D}$. The second step is when the previous $\mathcal{D}$ is used to estimate the position of our constructed compact ambiguity sets with the estimated nominal model $\bar{P}$ and the size ψ using Bayesian modelling. The final step is solving the Bellman equation and thus generate robust value functions and high-quality policies.

The choices of the specific methodologies employed in this work are presented here. Initially, we decided to adopt the RMDP formulation for our aim to appropriately model transition uncertainty via a constructed data-driven ambiguity set $\mathcal{P}$, rather than assuming a single known model. One of the main reasons is the focus on (s, a)-rectangular sets because with this structure we essentially keep robust Bellman planning tractable. One of the main novelties lies in the introduction of how we use the simplified Bootstrapped DQN within the ambiguity set construction and robust planning. Apart from the fact that it is a relatively known method and mainly remains computationally tractable, we believe the deep exploration will result in efficient data collection particularly in regions of the state space S that are poorly sampled. In addition, since the main objective is the construction of a compact data-driven ambiguity set, the collected uncertainty from this method should be accounted and positively influence the main elements of the set: the size and the location. Other ambiguity set construction methods mentioned in the literature, like other classical types of uncertainty sets or other RL algorithms, are not extensively implemented due to scope, apart from comparing some baselines.

4.3.1 Step 1: Bootstrapped Q-Value exploration

The first step and the specific methodological tool that we employ is partly inspired by Osband et al. (2016), borrowing the basic idea of the Bootstrapped DQN implementation and modifying it to a simplified variant, which essentially becomes an ensemble Q-learning variant and whose sole purpose is to gather diverse transition data. This will be specifically used for the adjustment of the location of the resulting ambiguity set, which is underexplored. Osband found that random initialisations of the network weights for each head were surprisingly effective by introducing in the end diversity, even if data is shared (Osband et al., 2016, 2019). Consequently, the result of our simplified approach is robust generated trajectories in Equation 4.5 and thus our primary goal of bootstrapped exploration will be to better estimate the transition probabilities, instead of the design for efficient exploration. Particularly, this will tackle the issue of limited data and allow the agent to further explore diverse trajectories that it would likely miss with other classic naive exploration methods, like ϵ-greedy (Sutton, 2018). This also helps optimise the location of the upcoming ambiguity sets.

Initially, the agent generates multiple Q-value function estimates like in Equation 4.3, which each represents a separate head of the neural network. Equation 4.8 is the Q-learning update for Bootstrapped DQN and for each bootstrapped head $k \in \{1, 2, \ldots, K\}$:

$$\theta_{i+1}^k \leftarrow \theta_i^k + \alpha\Big(r_i + \gamma \max_a Q_k(s_{i+1}, a; \bar{\theta}_i) - Q_k(s_i, a_i; \theta_i^k)\Big)\nabla_{\theta^k} Q_k(s_i, a_i; \theta_i^k) \quad (4.8)$$

where α is the learning rate and $r_i + \gamma \max_a Q_k(s_{i+1}, a; \bar{\theta}_i)$ is the target y_i^k for bootstrapped head k, while $\bar{\theta}_i$ is a fixed target network's parameter that is periodically updated every τ time steps for stable training (Osband et al., 2016; Zhou et al., 2020). $\nabla_{\theta^k} Q_k(s_i, a_i; \theta_i^k)$ is the gradient of the Q-function with respect to the parameters of head k. This equation defines how we adjust the Q-function parameters, θ_i, based on the observed experience that is continuously collected in Equation 4.5 for all episodes.

Instead of using a single Q-network, our approach uses a bootstrapped architec-

ture with a shared feature layers and K separate output heads, where each head $Q_k(s, a; \theta)$ acts as an independent Q-value estimator from equation 4.3. Initially, one of the bootstrapped Q-value heads is randomly selected from Uniform distribution, $k \sim \mathrm{U}\{1, \cdots, K\}$, meaning sampled uniformly with replacement (Neofytou et al., 2025; Owen and Eckles, 2012), and that specific head is then updated based on the transitions encountered during that episode. With that, the agent explored a wide and different range of pairs, and thus forming a more robust $\mathcal{D}$. Because the heads which are chosen randomly at the start differ, the agent observe different Q–value estimates and, over time, this creates slightly different parameter values in each head, which leads to diverse predictions.

Kindly note that for $K = 1$ we have the standard DQN, but with greedy action. Regarding the optimal choice of K, we would like to highlight that from Osband et al. (2016) the larger number of heads leads to faster learning, however the benefits of Bootstrapped DQN can be captured even in a small number of heads. In that work they decided to proceed with $K = 10$ heads which is sufficient for their experiment and for their aim. We acknowledge that the number of heads K is a tunable hyperparameter and its value is essentially reflected on the task and environment, while there is no universally optimal choice. We will further evaluate the number of heads at a later section.

A critical aspect in our modified approach is the parameter choice of mask p which is sampled from Bernoulli, $m_i \sim Bernoulli(p)$, resulting in either 0 or 1 for each data point with size the number of states s. The loss function and parameters of the deep network are updated for only data points with $m_i = 1$. Osband's approach used $p = 0.5$ to assign each data point with a specific head. In this case, only selected network weights, ones assigned to that data points, are updated from our deep network. For our experiments, we decided to adapt and proceed with 50% as well (Osband et al., 2016; Owen and Eckles, 2012), because it performed well. As a result, experience tuples are stored in the masked transition dataset, masked D, and the masks are uniformly sampled independently for training, while tuple points with $m_i = 0$ are ignored. Furthermore, it is also important to note the influence on the outcome of different initialisations of

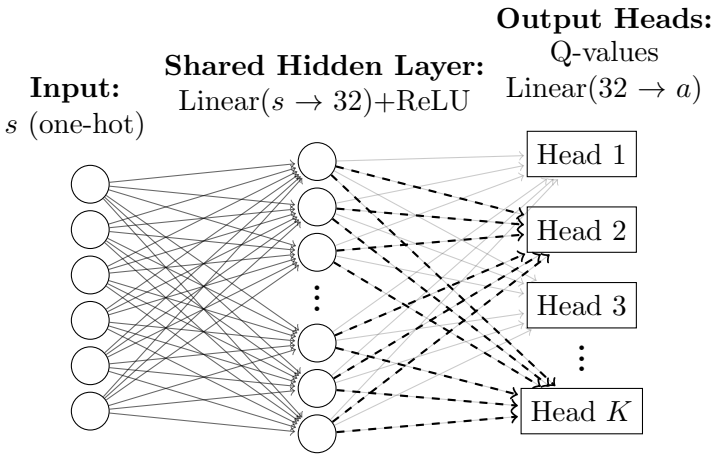


Figure 4.1: Bootstrapped DQN framework: Assuming $s = 6$ states as one-hot encoded input, and a shared hidden layer with 32 units and ReLU activation. The output involves K independent heads, each producing Q-values for $a = 2$ different actions. Initially, a single head is selected at random per episode and only the parameters of that specific head and the shared layers are updated, while the other heads remain frozen. The dashed black lines show the output heads that are active under the bootstrapped mechanism, for instance only Head 2 and Head K.

each head's network weights and the vital updates for the separate target networks every specific τ episodes.

Therefore, after the exploration phase, the agent has collected more robust transition tuples (s_i, a_i, s_{i+1}, m_i) from its previous interactions with the environment and are stored in the transition dataset $\mathcal{D}$, resulting in $\mathcal{D} = \{(s_i, a_i, s_{i+1}, m_i)\}_{i=1}^{N}$, following a similar thought process from Osband et al. (2016). In addition, the focus of this work is the ambiguity set construction rather than the optimisation of the Q-learning. However, we decided to employ a replay buffer $\mathcal{B}$ for Bootstrapped DQN, following the masked Bootstrapped DQN formulation of Osband et al. (2016). Specifically, each observed transition and its Bernoulli bootstrap mask are stored in a replay buffer $\mathcal{B}$. Following an initial period W, uniform minibatches are sampled from $\mathcal{B}$ for masked head-specific TD updates. The transition dataset $\mathcal{D}$ remains separate and records each genuine environmental transition. The algorithm of the simplified online Bootstrapped DQN variant exploration is given in Algorithm 4.

Furthermore, the architecture of the neural network employed for the smart robust exploration is given in Figure 4.1 for a one-hot encoded state input, a shared hidden

layer and the diverse output K heads with the actions. If $p = 0.5$, this indicates that each head is updated with 50% probability for each training sample, essentially, half of the heads are selected per sample on average. The dashed black lines provided imply the output heads that are selected for training, meaning the entire head is either active or inactive in the neural network. Some heads are completely skipped according to the previous Bernoulli mask p. The shared layers, input to hidden, are always updated.

We did not use the simplified bootstrapped DQN variant to directly find the policy, but instead we used this exploration method to get richer and more informative data, which will consequently result in a more informative estimate for the upcoming calculated robust policy. The bootstrap mechanism differs from classical bootstrap procedures such as basic or percentile intervals, which are methods for constructing confidence intervals. In contrast, Bootstrapped DQN uses an ensemble of value function heads to support exploration, as shown in Figure 4.1.

Algorithm 4 Exploration using Bootstrapped DQN

Input $\{Q_k\}_{k=1}^{K}$
Define $\mathcal{D} = \emptyset$, $\mathcal{B} = \emptyset$ and W threshold
Define bootstrapped mask p
for each episode **do**
    Choose k randomly from $k = \{1, \cdots, K\}$
    Receive initial state s_0
    for each step $i = 1, \ldots,$ final step **do**
        Choose action $a_i = \arg\max_a Q_k(s_i, a)$
        Receive s_{i+1}
        Receive r_i
        Sample mask $m_i^k \sim Bernoulli(p)$
        Store $(s_i, a_i, r_i, s_{i+1}, m_i)$ in both $\mathcal{D}$ and $\mathcal{B}$
        if $|\mathcal{B}| \geq W$ **then**
            Sample uniform minibatch $\mathcal{M} \subset \mathcal{B}$
            Perform TD updates for all heads using transitions $j \in \mathcal{M}$ for $m_j^k = 1$
        end if
    end for
end for
return $\mathcal{D}$

4.3.2 Step 2: Bayesian Ambiguity Set construction

In the second step we introduce the heart of RMDP, which is our data-driven ambiguity set construction. This is achieved by integrating the previous collected $\mathcal{D}$ in the environment from step 1. Specifically, from Algorithm 4, we now have more diverse and robust trajectories in $\mathcal{D} = \{(s_i, a_i, s_{i+1}, m_i)\}_{i=1}^{N}$ which will imply better posterior updates for transition probabilities. Then, inspired by Marek's Bayesian approach in Petrik and Russel (2019), we employ Bayesian modelling to generate compact ambiguity set for the transition probabilities.

Firstly, the resulting $\mathcal{D}$ will be used to calculate the nominal transitions probabilities $\bar{P}$ and consequently the location and position of our ambiguity sets. This estimated model accounts the uncertainty for model estimation error due to potential limited or non-robust data. Essentially, the nominal model $\bar{P}$ is the point estimate of the transition kernel obtained from the observed transition dataset $\mathcal{D}$ and it will be used as the centre of $\mathcal{P}_{s,a}$, while it is distinct from both the unknown true model $P^\star$ and the worst-case model $P \in \mathcal{P}$. To construct this, we rely on the posterior mean of the Dirichlet distribution, or empirical frequency of observed transitions, collected during step 1 for $\{(s_i, a_i, s_{i+1})\}_{i=1}^{N}$. The nominal transition probability is computed as:

$$\bar{P}(s' \mid s, a) = \frac{N(s, a, s') + \alpha_{s,a,s'}}{\sum_{j \in \mathcal{S}_{s,a}} (N(s, a, j) + \alpha_{s,a,j})}. \tag{4.9}$$

where $N(s, a, s')$ is the number of observed transitions from state s to state s' under action a in $\mathcal{D}$. We essentially sum over all possible outcomes and for cases where we have no observations, we will set the uniform distribution with $\alpha = 1$. The observed transitions in dataset $\mathcal{D}$ directly determine the ambiguity set centre through the posterior mean. Bootstrapped DQN does not alter the underlying transition kernel. Instead, it influences the state-action visitation frequencies and, consequently, the transition counts collected in $\mathcal{D}$. Therefore, it affects the estimated centre indirectly through the composition of the collected dataset.

Next, we will construct the size and radius of the ambiguity set, the uncertainty

around the estimated centre $\bar{P}$. This implies how far the true transition probabilities might deviate from $\bar{P}$ due to limited data. It is also a critical choice that essentially decides the trade-off between robustness and performance. As discussed, we assume a Dirichlet prior over the transition probabilities $P(s' \mid s, a)$ and for each action-state pair:

$$p_{s,a} \mid \mathcal{D} \sim \text{Dir}\left(\left\{\alpha_{s,a,s'} + N(s,a,s')\right\}_{s' \in \mathcal{S}_{s,a}}\right) \tag{4.10}$$

where $\alpha_{s,a,s'}$ is the previous non-informative parameter and prior. This is the posterior distribution and it is updated when we add the observed counts $n_{s,a}^{s'} = N(s,a,s')$ for every $s' \in \mathcal{S}_{s,a}$ to the corresponding Dirichlet prior parameter. We set the non-informative prior $a_{s,a,s'}$ to 1, a uniform prior, to avoid potential zero probabilities like before. As more data are collected, the posterior variance decreases and posterior samples concentrate around the posterior mean $\bar{P}$, which approaches P^* under posterior consistency. Equation 4.9 can also be calculated with the posterior mean of equation 4.10.

We will now proceed to use Bayesian credible regions for the final ambiguity set construction $\mathcal{P}_{\text{s,a}}$, which is defined as:

$$\mathcal{P}_{\text{s,a}} = \left\{P(s' \mid s, a) \in \Delta^S : \|P(s' \mid s, a) - \bar{P}(s' \mid s, a)\|_1 \leq \psi_{s,a}\right\} \tag{4.11}$$

where $\psi_{s,a}$ is the (dynamic) parameter that controls the tightness of the ambiguity set and Δ^S is the S-dimensional probability simplex, which implies that P are valid probability distributions with $P_i \geq 0$ and $\sum_{i=1}^{S} P_i = 1$. $\bar{P}$ is our best estimate of the true transition dynamics. We set our ambiguity sets using l_1 norm as a norm-distance, which is a standard and convenient way to solve efficiently. In general, the posterior distribution is used to optimise for the smallest ambiguity set around the nominal transition probability. Furthermore, the radius will be smaller due to equation 4.10 and we will be more confident about the true estimate. Specifically, from Marek's work in Petrik and Russel (2019), we calculate a set of $\psi_{s,a}$ and keep the smallest. Essentially, we choose the $[(1 - \frac{\delta}{L})M]^{th}$ smallest radius. M is the number of posterior draws and

$n_{s,a} = \sum_{s'} N(s,a,s')$ is the sample count, while $L = |\mathcal{S}||\mathcal{A}|$ and δ denotes the global posterior failure probability. We construct (s,a)-rectangular ambiguity sets as l_1-norm balls centred at the posterior mean transition vector $\bar{p}_{s,a}$. For every state-action pair, the radius is defined as:

$$\psi_{s,a} = \inf_{\psi \geq 0} \left\{ \psi : \mathbb{P}\left(\left\| p^{\star}_{s,a} - \bar{p}_{s,a} \right\|_1 \leq \psi \,\middle|\, \mathcal{D} \right) \geq 1 - \frac{\delta}{L} \right\} \tag{4.12}$$

In practice, the population radius in Equation 4.12 is approximated using M posterior draws $p^{(j)}_{s,a} \sim \text{Dirichlet}(\alpha_{s,a} + n_{s,a})$ from equation 4.10, and then $d_j = \|p^{(j)}_{s,a} - \bar{p}_{s,a}\|_1$ with $j = 1, \cdots, M$. After sorting the distances, we set the estimated radius $\psi_{s,a} = d_{(\lceil (1-\delta/L)M \rceil)}$.

The complete algorithm for the Bayesian ambiguity set construction is summarised in Algorithm 5 where the results influence the size and position of the ambiguity sets. Overall, the Dirichlet posterior ambiguity set construction adopted here follows established Bayesian RMDP approaches (Petrik and Russel, 2019; Petrik and Subramanian, 2016). Its role within this chapter is to convert the transition data collected through Bootstrapped DQN exploration in Step 1 into posterior-calibrated state-action ambiguity sets for robust planning. The contribution lies in this specific integration and its evaluation.

Algorithm 5 Bayesian Ambiguity Set Construction

Input Transition Dataset $\mathcal{D}$, Prior Dirichlet parameters $\alpha^{s'}_{s,a}$, confidence level δ, posterior draw count M
Initialize $d = \emptyset$, $X = \emptyset$ and $\alpha_{s,a} = 1$
Count $n_i \in N(s,a,s)$ in $\mathcal{D}$
for $i = 1$ to M **do**
    Sample $X_i \sim \text{Dir}(\alpha^{s_1}_{s,a} + n^{s_1}_{s,a}, \alpha^{s_2}_{s,a} + n^{s_2}_{s,a}, \ldots, \alpha^{s_{|\mathcal{S}|}}_{s,a} + n^{s_{|\mathcal{S}|}}_{s,a})$
    Append$X \leftarrow X_i$
end for
Calculate $\bar{P}_{s,a} \leftarrow \frac{1}{M} \sum_{i=1}^{M} X_i$
for $i = 1$ to M **do**
    Calculate $d_i \leftarrow \|X_i - \bar{P}_{s,a}\|_1$
    Append $d \leftarrow d_i$ (sort-ascending)
end for
Calculate $\psi_{s,a} \leftarrow d_{\left(\lceil (1-\frac{\delta}{L})M \rceil\right)}$
return $\bar{P}_{s,a}$ and $\psi_{s,a}$

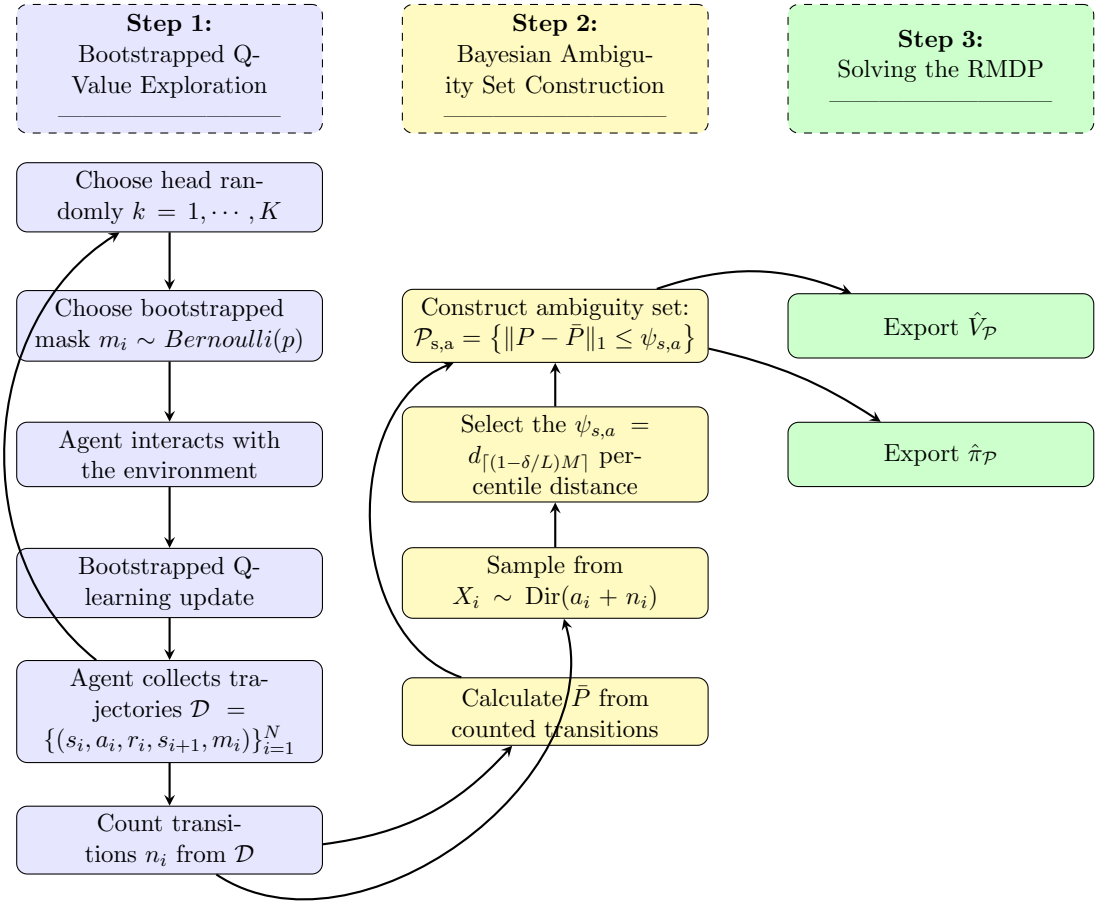


Figure 4.2: Flowchart of our framework

4.3.3 Step 3: Solving the RMDP

The third and final step of our framework involves solving the RMDP. This implies solving the robust bellman equation in Equation 4.7 with the calculated $\mathcal{P}_{\text{s,a}}$ from previous step, which involves the estimated size and location from previous steps. The formulation of the ambiguity set is tractable with the l_1 norm and thus we solve with Linear Programming. The results are: the robust estimated value function $\hat{V}_{\mathcal{P}}$ and the estimated policy $\hat{\pi}_{\mathcal{P}}$. Consequently, the resulting estimated policy from our methodology reflects its robustness under uncertainty when we do not know the exact dynamics in an environment. Specifically, the RMDP is solved using robust value iteration and, for each state-action pair, the inner worst-case expectation over $\mathcal{P}_{s,a}$ is solved as a linear

program over the globally calibrated l_1 ambiguity set in Equation 4.11. The holistic framework is given in Figure 4.2. The following subsection establishes the theoretical properties of this robust planning stage.

4.3.4 Theoretical properties

The following results concern the robust planning stage after the transition dataset $\mathcal{D}$ has been collected. Let $L = |\mathcal{S}||\mathcal{A}|$ denote the number of state-action pairs. We assume finite state and action spaces, bounded rewards, and $\gamma \in [0, 1)$.

For every (s, a), the radius in Equation 4.12 in Step 2 is calibrated such that

$$\mathbb{P}\left(p^\star_{s,a} \notin \mathcal{P}_{s,a} \,\middle|\, \mathcal{D}\right) \leq \frac{\delta}{L}.$$

From equation 4.4, the complete ambiguity set remains (s, a)-rectangular:

$$\mathcal{P} = \prod_{s \in \mathcal{S}} \prod_{a \in \mathcal{A}} \mathcal{P}_{s,a}.$$

Lemma 1 *The globally calibrated ambiguity set satisfies*

$$\mathbb{P}\left(P^\star \in \mathcal{P} \,|\, \mathcal{D}\right) \geq 1 - \delta.$$

For each (s, a), we define the failure event

$$E_{s,a} = \left\{p^\star_{s,a} \notin \mathcal{P}_{s,a}\right\}.$$

Since

$$\{P^\star \notin \mathcal{P}\} = \bigcup_{s \in \mathcal{S}} \bigcup_{a \in \mathcal{A}} E_{s,a},$$

the union bound gives

$$\begin{aligned}\mathbb{P}\left(P^\star \notin \mathcal{P} \,|\, \mathcal{D}\right) &\leq \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \mathbb{P}(E_{s,a} \mid \mathcal{D}) \\ &\leq L \frac{\delta}{L} = \delta.\end{aligned}$$

Taking complements proves the result.

Since $\mathcal{P}$ is (s,a)-rectangular and $\gamma < 1$, the robust Bellman operator is a γ-contraction under the supremum norm. Therefore, robust value iteration converges to its unique fixed point (Iyengar, 2005; Nilim and El Ghaoui, 2005).

Theorem 1 *Let $\widehat{\pi}_{\mathcal{P}}$ be the robust policy obtained by solving the RMDP, and let $\widehat{V}_{\mathcal{P}}^{\widehat{\pi}_{\mathcal{P}}}$ denote its robust value. Then*

$$\mathbb{P}\left(\widehat{V}_{\mathcal{P}}^{\widehat{\pi}_{\mathcal{P}}}(s) \le V_{P^\star}^{\widehat{\pi}_{\mathcal{P}}}(s), \quad \forall s \in \mathcal{S} \,\middle|\, \mathcal{D}\right) \ge 1-\delta$$

We consider the event $\{P^\star \in \mathcal{P}\}$ and the true transition vector $p^\star_{s,a}$ belongs to every corresponding row set $\mathcal{P}_{s,a}$. Therefore, for any policy π and value vector v,

$$\big(\widehat{T}_{\mathcal{P}}^{\pi} v\big)(s) = \min_{p \in \mathcal{P}_{s,\pi(s)}} p^{\mathsf{T}}\big(r_{s,\pi(s)} + \gamma v\big) \le \big(T_{P^\star}^{\pi} v\big)(s)$$

Monotonicity and contraction of the Bellman operators consequently imply

$$\widehat{V}_{\mathcal{P}}^{\pi}(s) \le V_{P^\star}^{\pi}(s), \qquad \forall s \in \mathcal{S}.$$

By applying this result to $\widehat{\pi}_{\mathcal{P}}$ and using Lemma 1 proves the theorem. ■

If $\pi^\star$ denotes the optimal policy under the true dynamics, then, on the same coverage event,

$$\widehat{V}_{\mathcal{P}}^{\widehat{\pi}_{\mathcal{P}}}(s) \le V_{P^\star}^{\widehat{\pi}_{\mathcal{P}}}(s) \le V_{P^\star}^{\pi^\star}(s)$$

The following results use the exact posterior radius $\psi_{s,a}$. From equation 4.12, let M denote the number of posterior draws and define

$$d_j = \left\| p_{s,a}^{(j)} - \bar{p}_{s,a} \right\|_1, \qquad j = 1, \dots, M.$$

After sorting the distances, the implemented radius is $\widehat{\psi}_{s,a} = d_{(\lceil (1-\frac{\delta}{L})M \rceil)}$. Thus, $\widehat{\psi}_{s,a}$ is a Monte Carlo approximation, Sample Average Approximation (SAA), of the exact posterior quantile. The theoretical guarantee applies exactly to $\psi_{s,a}$ and to the implemented radius up to posterior quantile-estimation error, which is treated as

negligible for sufficiently large M (Petrik and Russel, 2019).

These results justify the Bayesian ambiguity set and robust planning stages, but they do not establish theoretical superiority of Bootstrapped DQN over ϵ-greedy exploration.

4.4 Experiments

In this section, we present two experiments to evaluate the performance of our holistic method. First, we demonstrate the improved solution quality of our approach with a standard and slightly modified RiverSwim domain, which is a low-dimensional controlled environment. Subsequently, we test the proposed method with a modified inventory optimisation experiment with some uncertainty involved. Simulation is standard in RMDP, thus we assess the proposed method in those simulated benchmarks, while also evaluating how our RMDP results perform with uncertainty from lack of data and uncertainty from model parameters. In addition, each $\mathcal{P}_{s,a}$ uses the posterior failure $\frac{\delta}{|\mathcal{S}||\mathcal{A}|}$ and therefore targets posterior credibility $\frac{\delta}{|\mathcal{S}||\mathcal{A}|}$. By the union bound, their Cartesian product $\mathcal{P}$ targets joint posterior coverage of at least $1-\delta$. The target is the construction of the ambiguity set $\mathcal{P}$ which represents uncertainty about which transition model/kernel P is correct because the available dataset is finite. For every method, we make the following evaluation:

- **Robust certificate:** $A = \min_{P \in \mathcal{P}} V_P^{\widehat{\pi}}(s_0)$

  which is the worst-case return of the learned robust policy $\widehat{\pi}$ over the constructed ambiguity set.

- **True model policy value:** $B = V_{P^\star}^{\widehat{\pi}}(s_0)$

  which evaluates the learned policy $\widehat{\pi}$ under the true transition dynamics $P^\star$.

- **Oracle optimal value:** $C = V_{P^\star}^{\pi^\star}(s_0)$

  which is the optimal simulator performance benchmark. It is the same for every method because $\pi^\star$ is the optimal policy under $P^\star$.

Additionally, we report the following gaps:

- **Robust return error:** $|C - A| = |V_{P^\star}^{\pi^\star}(s_0) - \min_{P \in \mathcal{P}} V_P^{\widehat{\pi}}(s_0)|$

  which is the absolute difference between the robust certificate and the optimal return, following the absolute error evaluation used by Petrik and Russel (2019).

- **Certificate gap:** $|B - A| = |V_{P^\star}^{\widehat{\pi}}(s_0) - \min_{P \in \mathcal{P}} V_P^{\widehat{\pi}}(s_0)|$

  with a value near zero indicates a tight certificate.

- **Policy optimal gap:** $|C - B| = |V_{P^\star}^{\pi^\star}(s_0) - V_{P^\star}^{\widehat{\pi}}(s_0)|$

  which measures the suboptimality of the learned policy under the true dynamics. Smaller values indicate that $\widehat{\pi}$ is closer to the true model optimal policy.

Whenever $P^\star \in \mathcal{P}$, the theoretical coverage result implies $A \leq B \leq C$ and hence $C - A = (B - A) + (C - B)$. All experiments were run on a university laptop with an 11th Gen Intel(R) Core(TM) i5-1145G7 @ 2.60GHz and 16GB RAM.

4.4.1 RiverSwim domain

We first introduce a simple RiverSwim domain (Petrik and Russel, 2019; Strehl and Littman, 2008), a widely used benchmark. The specific modified problem involves 6 states, starting from state 0 and the aim is to reach state 5. There are 2 actions, stay at the same state or swim. In this setting, uncertainty arises from lack of data, limited and non-robust observed transitions, to solve the RMDP. Particularly, because we have only limited samples, the agent is not sure about the exact transition distribution P^*. Thus we need to gather enough robust observations in $\mathcal{D}$ in order to properly estimate the transition probability $\bar{P}$ and construct our ambiguity set around this and account for the uncertainty.

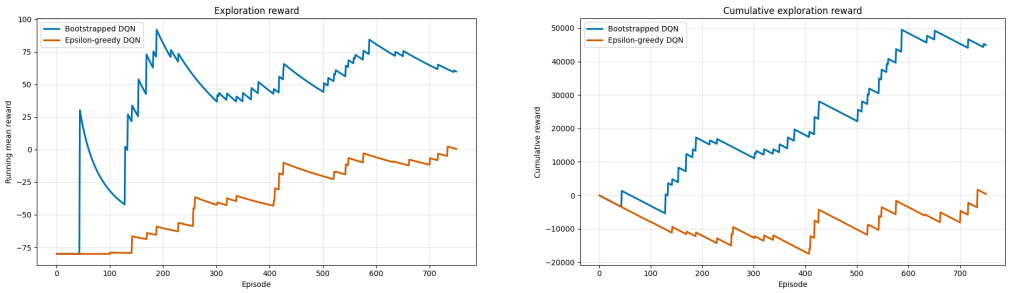
The rewards are -1 for states s_0, s_1, s_2, s_3, s_4 and +100 for the final state s_5. The initial state is s_0. We also generated a synthetic dataset from the ground truth for comparison following "true" transition probabilities for Riverswim, where we use P^* to simulate next state transitions while the agent does not know or use it in the learning process. Specifically, for the s_0 we have 95% to stay and 5% to swim, and for s_1, s_2, s_3, s_4

Table 4.1: Hyperparameters used for RiverSwim experiment

Parameter	**Value**
Train iterations/episodes	750
Steps per iteration/episode	80
Bootstrapped heads K	10
Episodes to update model τ	100
Mask bootstrapped for Bernoulli p	0.5
Non-informative prior a_i	1
Discount factor γ	0.99
Learning rate	0.001
Hidden layer	(32)
Optimiser	Adam
Activation	ReLU
Confidence level for Bayesian	95%
ϵ-greedy Q-learning	0.2

we have 90% to stay, 5% to swim and 5% to slip and start from the start. The hyperparameters for the RiverSwim experiment are provided in Table 4.1. We also compare with Hoeffding inequality calculation $\psi_{Hoeffding} = \sqrt{\frac{2}{N} * log(\frac{SA2^S}{\delta})}$ (Petrik and Russel, 2019). Lastly, the number of bootstrapped heads, K, is set to $K = 10$ for our method in this experiment.

For the exploration, Figure 4.3a presents the mean reward of each method per episode for a single run. It indicates that the ϵ-greedy exploration, which implies a single head DQN baseline with ϵ-greedy exploration, shows slower and less stable improvement with early fluctuations. On the other hand, bootstrapped exploration, which implies actions that are chosen greedily with respect to that Bootstrapped DQN exploration head, shows more consistent learning and reaches significantly higher rewards faster. This is mainly due to its more diverse exploration strategy, greedy with K heads for Bootstrapped DQN vs ϵ-greedy with standard DQN. Figure 4.3b displays the cumulative reward of each method, where Bootstrapped DQN significantly outperforms the ϵ-greedy DQN method over time. Initially, both methods start with negative rewards, but Bootstrapped DQN indicates a faster improvement in cumulative reward, reaching



(a) (b)

Figure 4.3: Exploration for RiverSwim: (a) Mean reward for each exploration method and (b) Cumulative reward for each exploration method

a higher reward than ϵ-greedy exploration. We expect that our method will lead to a more accurate estimation of the key elements of the upcoming ambiguity set.

After the exploration stage, we have noticed that our Bayesian method constructs ψ values that are not similar to all states-actions pairs, and neither excessively large nor overly conservative, compared to other approaches. It essentially decides the states in which to act less conservatively and those where a more conservative and pessimistic approach is needed, like states s_4 and s_5. Furthermore, we can observe in Table 4.2 the individual exported centres $\bar{P}$ of each individual exploration method, indicating their preferences at (s,a) pairs, and both close to the optimal P^*. As expected, the exploration method heavily influences the outcome, and will further affect the resulting robust value function for each state and exported final policy. Specifically, we noticed that some (s,a) pairs are quite similar, some demonstrate small differences, while others show more noticeable differences. We interpret these results as follows: the exploration methods behave similarly in the initial states, but the performance gap becomes higher as the number of state increases. We believe that this suggests that the underlying algorithmic architecture causes increasingly different estimates in later parts of the state space. We conclude that the exploration method indeed affects primarily the main elements of the ambiguity set, like the ψ value as radius and, therefore, $\bar{P}$ as centres in Table 4.2. Once again, we anticipate that this will affect the final ambiguity set construction, while one of our objectives and research questions was to examine whether, and how, it affects the quality of the final solution.

Table 4.2: Comparison of the ambiguity set centres $\bar{P}$ obtained using Bootstrapped DQN and ϵ-greedy exploration. The columns represent the next state.

State	Action	$\bar{P}_{\text{boot}}$ (Bootstrapped DQN)						$\bar{P}_{\epsilon}$ (ϵ-greedy)					
		s_0	s_1	s_2	s_3	s_4	s_5	s_0	s_1	s_2	s_3	s_4	s_5
s_0	$a=0$	1.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000
	$a=1$	0.951	0.049	0.000	0.000	0.000	0.000	0.950	0.050	0.000	0.000	0.000	0.000
s_1	$a=0$	0.000	1.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000
	$a=1$	0.050	0.899	0.051	0.000	0.000	0.000	0.051	0.901	0.048	0.000	0.000	0.000
s_2	$a=0$	0.045	0.045	0.773	0.045	0.045	0.045	0.000	0.000	0.999	0.000	0.000	0.000
	$a=1$	0.048	0.000	0.902	0.050	0.000	0.000	0.051	0.000	0.901	0.048	0.000	0.000
s_3	$a=0$	0.017	0.017	0.017	0.914	0.017	0.017	0.001	0.001	0.001	0.997	0.001	0.000
	$a=1$	0.050	0.000	0.000	0.899	0.051	0.000	0.047	0.000	0.000	0.898	0.054	0.000
s_4	$a=0$	0.167	0.167	0.167	0.167	0.167	0.167	0.001	0.001	0.001	0.001	0.996	0.001
	$a=1$	0.050	0.000	0.000	0.000	0.900	0.049	0.053	0.000	0.000	0.000	0.889	0.057
s_5	$a=0$	0.167	0.167	0.167	0.167	0.167	0.167	0.000	0.000	0.000	0.000	0.000	0.998
	$a=1$	0.000	0.000	0.000	0.000	0.000	0.999	0.003	0.003	0.003	0.003	0.003	0.984

Going into details, Table 4.3 shows the robust planning results from the initial state s_0 (Nilim and El Ghaoui, 2005). As we discussed before, the robust certificate A represents the worst-case expected return, if we start from there and follow the robust policy, and it is obtained when the learned robust policy is evaluated over its constructed ambiguity set. The table reports the mean and sample standard deviation over 20 independent runs.

We observe that, among the learned methods, Bootstrapped DQN with the globally calibrated Bayesian radius achieves the highest mean robust certificate, $A = 94.10 \pm 85.27$, compared with 25.97 ± 62.87 for the corresponding ϵ-greedy method. More importantly, its true-model policy value is $B = 959.44 \pm 0.00$, equal to the oracle value $C = 959.44$ in every one of the 20 independent runs. Consequently, it has the lowest robust return error, but also a zero policy-oracle gap. This implies that it consistently recovers a policy that is optimal under the true RiverSwim dynamics, $P^{\star}$, in this experiment. In comparison, the Bayesian ϵ-greedy method obtains $B = 906.47 \pm 236.90$ and a non-zero mean policy-oracle gap of 52.97 ± 236.90, which implies a less consistent policy across seeds, which we will visually show later.

Table 4.3: RiverSwim: comparison of robust planning methods for 750 training episodes over 20 independent runs. Entries report the mean ± sample standard deviation. Bold entries identify the best learned method for the corresponding column. As mentioned: $A = \min_{P \in \mathcal{P}} V_P^{\widehat{\pi}}(s_0)$ and $B = V_{P^\star}^{\widehat{\pi}}(s_0)$.

Method	**Robust certificate** A	**True-policy value** B	**Robust-return error** $\lvert C - A \rvert$	**Certificate gap** $\lvert B - A \rvert$	**Policy-oracle gap** $\lvert C - B \rvert$
Bootstrapped DQN with $\psi = 0.1$	-100.00 ± 0.00	-100.00 ± 0.00	1059.44 ± 0.00	0.00 ± 0.00	1059.44 ± 0.00
Bootstrapped DQN with $\psi = 0.3$	-100.00 ± 0.00	-100.00 ± 0.00	1059.44 ± 0.00	0.00 ± 0.00	1059.44 ± 0.00
Bootstrapped DQN with global Bayesian ψ	$\mathbf{94.10 \pm 85.27}$	$\mathbf{959.44 \pm 0.00}$	$\mathbf{865.34 \pm 85.27}$	$\mathbf{865.34 \pm 85.27}$	$\mathbf{0.00 \pm 0.00}$
Bootstrapped DQN with Hoeffding ψ	-99.84 ± 0.72	-47.03 ± 236.90	1059.28 ± 0.72	52.81 ± 236.18	1006.47 ± 236.90
DQN ϵ-greedy with $\psi = 0.1$	-100.00 ± 0.00	-47.03 ± 236.90	1059.44 ± 0.00	52.97 ± 236.90	1006.47 ± 236.90
DQN ϵ-greedy with $\psi = 0.3$	-100.00 ± 0.00	-100.00 ± 0.00	1059.44 ± 0.00	0.00 ± 0.00	1059.44 ± 0.00
DQN ϵ-greedy with global Bayesian ψ	25.97 ± 62.87	906.47 ± 236.90	933.47 ± 62.87	880.50 ± 214.53	52.97 ± 236.90
DQN ϵ-greedy with Hoeffding ψ	-100.00 ± 0.00	-100.00 ± 0.00	1059.44 ± 0.00	0.00 ± 0.00	1059.44 ± 0.00

Simulator benchmark: $C = V^{\pi^\star}(P^\star, s_0) = 959.44 \pm 0.00$

In addition, the worst case expected return, A, of the proposed method remains substantially below its true-model policy value, B, producing a certificate gap of 865.34 ± 85.27. This does not show poor true-environment performance: it shows that the ambiguity set still contains some adverse transition models and, therefore, produces a conservative worst-case return. The fixed radius and Hoeffding alternatives generally return robust certificates close to -100 and substantially lower true-model values. Moreover, the zero certificate gaps of some fixed-radius methods do not imply superior performance, because both their robust certificates and true-model policy values are approximately -100. We observe that the results, for this specific experiment, provide empirical evidence that combining Bootstrapped DQN exploration with the globally calibrated Bayesian ambiguity set produces the strongest worst-case return, but also the most consistent true-model policy performance among the evaluated methods, while also outperforming in all errors and gaps.

Furthermore, Figure 4.4 visualizes the previous exported outcomes, specifically the means and 95% confidence intervals for each method. As we described before, we observe that, as additional exploration data are collected, our method achieves the highest robust certificate A and true-policy B. It consequently produces the smallest error in gap $|C - A|$ and gap $|C - B|$. The ϵ-greedy Bayesian method also improves

substantially, but presents a lower final worst-case return.

The learned robust policies are presented in Figure 4.5. Specifically, Figure 4.5a shows the policy obtained for one seed with actions 0 or 1, while Figure 4.5b reports the fraction of the 20 runs for selecting action 1 in each state. Bootstrapped DQN with Bayesian ψ recovered the oracle policy $[s_0 = 1, s_1 = 1, s_2 = 1, s_3 = 1, s_4 = 1, s_5 = 0]$, which is the optimal for our Riverswim experiment, in the single seed in Figure 4.5a, but also in all 20 runs with 100% success in Figure 4.5b. The ϵ-greedy Bayesian approach was also close to the oracle policy, selecting the correct action 1 in 95% of the 20 runs, meaning in 19 out of the 20 seeds, in several states. However, the fixed-radius and Hoeffding variants, even though we observe some effort, they were considerably less consistent.

As an additional exploration evaluation, we increased the number of episodes from 750 to 1500. Table 4.4 reports the across-run means and 95% Student-t confidence intervals over 20 independent seeds. We observe that, as the number of episodes increases, other conservative approaches attempt to find an even better policy and higher robust values, in contrast to Table 4.3 where they just demonstrated some initial effort. On the other hand, both global Bayesian approaches achieved the oracle true-model value. However, once again, Bootstrapped DQN achieves a higher robust certificate and a smaller robust return error than ϵ-greedy DQN. This implies a tighter and less conservative certificate as more transition data are collected.

Table 4.4: RiverSwim: comparison of robust planning methods for 1500 training episodes over 20 independent runs. Entries report the mean [95% Student-t CI]. Bold entries identify the best learned method for the corresponding column.

Method	**Certificate** A	**True value** B	$\lvert C - A \rvert$	$\lvert B - A \rvert$	$\lvert C - B \rvert$
Bootstrapped DQN with $\psi = 0.1$	$-100.00\,[-100.00, -100.00]$	$-100.00\,[-100.00, -100.00]$	$1059.44\,[1059.44, 1059.44]$	$0.00\,[0.00, 0.00]$	$1059.44\,[1059.44, 1059.44]$
Bootstrapped DQN with $\psi = 0.3$	$-100.00\,[-100.00, -100.00]$	$-100.00\,[-100.00, -100.00]$	$1059.44\,[1059.44, 1059.44]$	$0.00\,[0.00, 0.00]$	$1059.44\,[1059.44, 1059.44]$
Bootstrapped DQN with global Bayesian ψ	$\mathbf{277.66\,[244.86, 310.46]}$	$\mathbf{959.44\,[959.44, 959.44]}$	$\mathbf{681.78\,[648.98, 714.58]}$	$\mathbf{681.78\,[648.98, 714.58]}$	$\mathbf{0.00\,[0.00, 0.00]}$
Bootstrapped DQN with Hoeffding ψ	$-64.51\,[-77.97, -51.04]$	$694.58\,[474.30, 914.86]$	$1023.95\,[1010.48, 1037.41]$	$759.08\,[548.44, 969.72]$	$264.86\,[44.58, 485.14]$
DQN ϵ-greedy with $\psi = 0.1$	$-100.00\,[-100.00, -100.00]$	$-100.00\,[-100.00, -100.00]$	$1059.44\,[1059.44, 1059.44]$	$0.00\,[0.00, 0.00]$	$1059.44\,[1059.44, 1059.44]$
DQN ϵ-greedy with $\psi = 0.3$	$-100.00\,[-100.00, -100.00]$	$-100.00\,[-100.00, -100.00]$	$1059.44\,[1059.44, 1059.44]$	$0.00\,[0.00, 0.00]$	$1059.44\,[1059.44, 1059.44]$
DQN ϵ-greedy with global Bayesian ψ	$181.32\,[155.14, 207.49]$	$\mathbf{959.44\,[959.44, 959.44]}$	$778.13\,[751.95, 804.30]$	$778.13\,[751.95, 804.30]$	$\mathbf{0.00\,[0.00, 0.00]}$
DQN ϵ-greedy with Hoeffding ψ	$-99.44\,[-100.28, -98.59]$	$5.95\,[-146.67, 158.56]$	$1058.88\,[1058.03, 1059.72]$	$105.38\,[-46.42, 257.18]$	$983.50\,[800.88, 1166.11]$

Furthermore, we will provide the sensitivity analysis for the key hyperparameter of

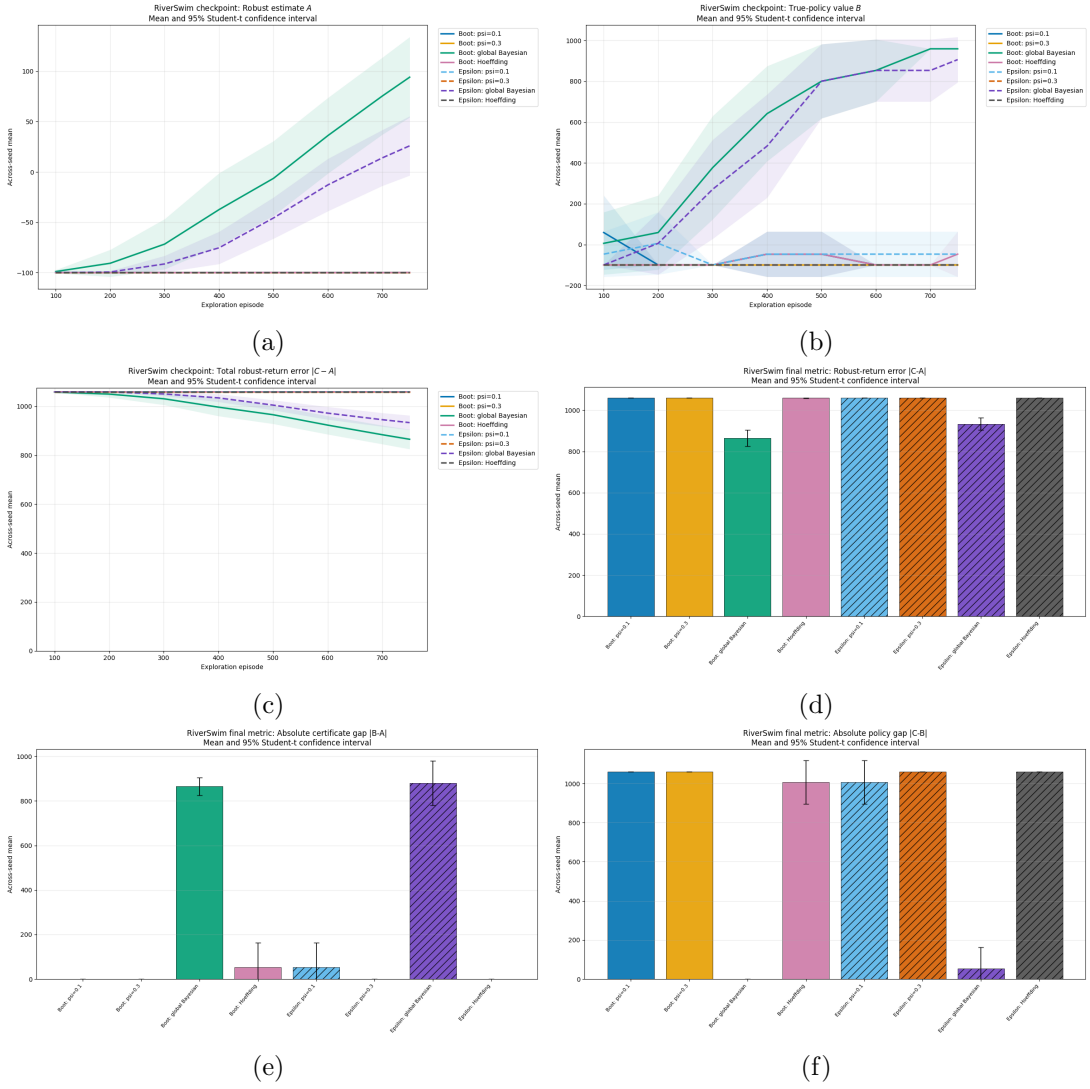
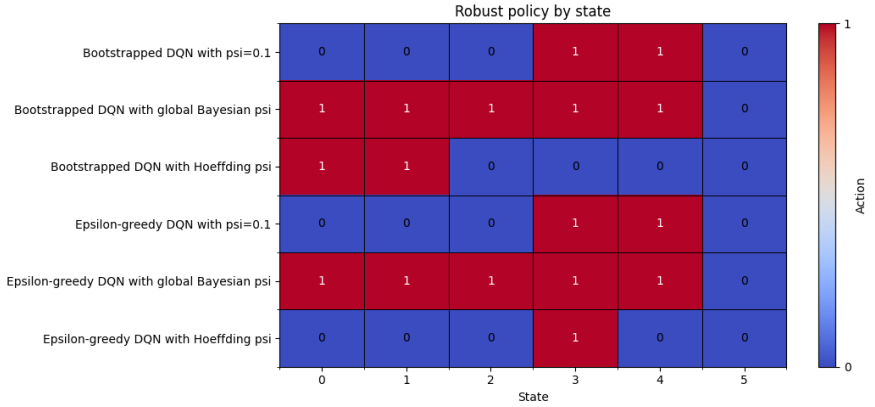
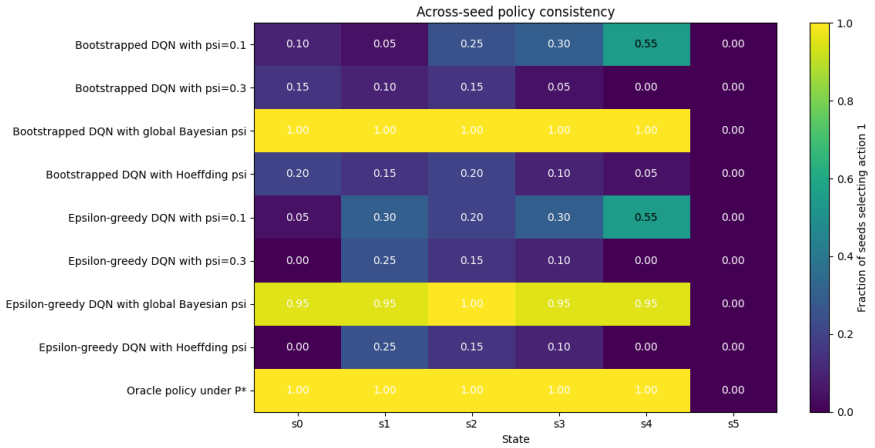


Figure 4.4: RiverSwim results over 20 independent runs, showing means and 95% Student-t confidence intervals as shaded regions in (a)–(c) and error bars in (d)–(f). Specifically: (a) the robust certificate A, (b) the true-model value B of the learned robust policy, (c) its evolution as additional exploration data are collected, (d) its final value for each method. Lower values indicate that the robust certificate is closer to the oracle return, (e) the absolute certificate gap $|B - A|$, and (f) the absolute policy-oracle gap $|C - B|$. Higher values are preferable in (a)-(b), while lower values are preferable in (c)-(f).



(a)



(b)

Figure 4.5: Policy schedule for RiverSwim after 750 training episodes: (a) policy for a single seed and (b) policy consistency with percentage of success for taking action 1, across 20 seeds.

our method, the number of heads K, as a supporting evidence. Table 4.5 provides an empirical stability and sensitivity analysis of the Bootstrapped DQN procedure across different numbers of heads. In addition, this will be averaged over 10 new, unseen, independent random seeds for each method. In column "Method", when we are talking about "Bootstrapped DQN" we are essentially describing a greedy exploration for K heads, by choosing one head randomly at the start of the episode, then act greedily with respect to that head during the whole episode, as described in Algorithm 4. The comparison other method was a single head DQN baseline with ϵ-greedy exploration. We decided to evaluate only over 10 independent random seeds due to the high com-

Table 4.5: RiverSwim: Sensitivity Analysis – Evaluation of different number of K (over 10 independent random seeds for each method) with Bayesian ψ. Each independent run was executed over 750 episodes and for each episode 80 steps.

Number of K	**Method**	***Robust Value*** *($V(s_0)$)*		**Mean Time (s)**
		Mean ± St. Deviation	**95% CI**	
2	Bootstrapped DQN	101.21 ± 86.99	[38.98, 163.44]	652
3	Bootstrapped DQN	115.58 ± 61.62	[71.50, 159.66]	1023
4	Bootstrapped DQN	117.91 ± 69.64	[68.09, 167.73]	1173
5	Bootstrapped DQN	109.64 ± 51.24	[72.99, 146.29]	1324
7	Bootstrapped DQN	95.97 ± 50.78	[59.64, 132.30]	1625
10	Bootstrapped DQN	112.94 ± 61.56	[68.90, 156.98]	2125
15	Bootstrapped DQN	104.23 ± 72.45	[52.40, 156.06]	2829
–	DQN with ϵ-greedy	20.44 ± 45.33	[-11.99, 52.87]	530

putational time, while we also assess over new and unseen runs rather than a single random outcome. We initially provide the number of heads K and each method used along with employed Bayesian ψ as radius. Then, we evaluate the mean ± the sample standard deviation of the worst-case return of robust value, over those 10 unseen independent random seeds. We also evaluate the 95% Confidence Interval (CI) of the 10 new independent random runs. Lastly, we provide the average time for each method to be executed. We need to mentioned that we kept all the others hyperparameters stable with 750 episodes and 80 steps per episode. The results show that every Bootstrapped DQN variant has a higher mean $V(s_0)$ than DQN with ϵ-greedy. This is consistent with one of our aims and supports the hypothesis that deep exploration improves the transition data $\mathcal{D}$, and thus allows a less overly conservative ambiguity set construction and improved outcome. Moreover, evaluating the $K = 10$ configuration on 10 additional independent seeds produced a similar mean robust certificate and comparable across-seed variability to the 20-seed results in Table 4.3 and Figure 4.4, providing an additional robustness check. A visual representation of the different number of heads

is presented in Figure 4.6.

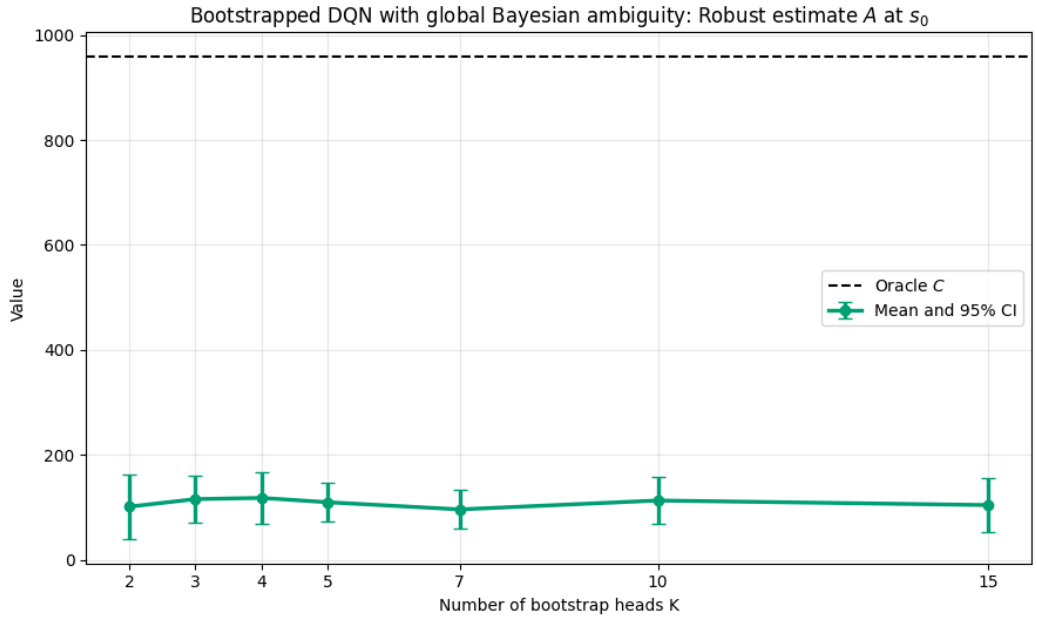


Figure 4.6: RiverSwim sensitivity of the robust certificate $A = \min_{P \in \mathcal{P}} V_P^{\widehat{\pi}}(s_0)$ to the number of Bootstrapped DQN heads K after 750 exploration episodes. Markers show the means over 10 independent runs, error bars show 95% Student-t confidence intervals, and the dashed line indicates the previous oracle value $C = 959.44$.

However, the relatively large sample standard deviations and overlapping 95% confidence intervals, mainly due to the low number of training episodes, imply some substantial across-seed variability and show that, with ten runs, the results do not clearly distinguish one value of K as superior. Such sensitivity to random seeds is well documented in DRL (Henderson et al., 2018), while reproducible comparisons across runs are also emphasised in RRL benchmarking (Zouitine et al., 2024). Accordingly, our conclusions are drawn based on these aggregated results rather than individual runs. The mean robust value does not improve monotonically with K, whereas computational time increases substantially as more heads are added. Preliminary trials with $K = 50$ required approximately 7850 seconds without an evident performance improvement. These results are treated only as a computational observation, while the findings are consistent with Osband et al. (2016), who observed that a small-to-moderate number of heads can already capture most of the benefit, and thus a moderate K=10 is selected

for practicality.

Due to the fact that the transition observations in this chapter are categorical tuples, explicit outlier removal from $\mathcal{D}$ is not applied since a rarely observed transition may still correspond to a genuine low-probability transition. Therefore, removing such observations could bias the estimated transition and reduce the ambiguity set. As shown, sensitivity is instead considered through multiple independent random seeds in Table 4.3, standard deviation, and CI, while we need to highlight that the initial choice of the confidence level for Bayesian with 95% also affects the outcome.

We demonstrated that, under the tested RiverSwim setting, ϵ-greedy exploration could eventually recover the oracle policy as the number of episodes increased, but generally produced far lower robust certificates A, higher gaps, and more conservative solutions. In contrast, Bootstrapped DQN exploration produced higher robust certificates and more consistently recovered the oracle policy across seeds. The globally calibrated Bayesian radius ψ also influenced the performance. Bootstrapped DQN encourages deeper exploration and changes the frequency of various state-action visits, producing more informative transition counts. Frequently visited (s, a) pairs consequently obtain more concentrated posterior estimates and smaller radius, which reduces unnecessary pessimism, while poorly sampled pairs encouraged larger radius and stronger protection. This behaviour explains the improved worst-case returns of our method, observed in RiverSwim, compared with fixed-radius or other concentration-based alternatives. However, this trade-off is empirical and is not guaranteed in every environment and also generates a handful of limitations that we will expand at a later section.

4.4.2 Inventory optimisation domain

We next proceed with an inventory optimisation case, an extension of a simplified version from Petrik and Russel (2019). This is an area with a rich history of stochastic and RO applications, see Attila et al. (2021). In this modified setting, we have 20 states, $s_0, \cdots, s_{19}$, representing discrete inventory levels, $s \in 0, 1, \ldots, 19$, where $s = 0$ corresponds to no available stock and $s = 19$ implies the full capacity. The action space involves only four actions, $a_0, \cdots, a_3$, which $a \in \{0, 1, 2, 3\}$ implies the number

of product units ordered.

After selecting action a_t in state s_t, the post-order inventory is $x_t = \min\{s_t + a_t, 19\}$. This inventory subsequently faces stochastic demand D_t. Consequently, the next state is $s_{t+1} = \max\{x_t - D_t, 0\}$. Demand is generated by rounding a Normal random variable and is provided below:

$$D_t = \max\{0, \text{round}(Z_t)\}, \qquad Z_t \sim \mathcal{N}(2, 2), \tag{4.13}$$

where the distribution's mean is 2 and its variance is 2, while we are essentially generating non-negative demand for each t. The simulator transition model $P^\star$ is constructed using a fixed Monte Carlo approximation based on 10,000 demand draws and is held constant across exploration seeds. Once again, the learning methods are not given $P^\star$, instead, they observe sampled transitions and estimate the transition probabilities from finite data. The feasible next-state, $0 \leq s_{t+1} \leq x_t$, is assumed to be known from the inventory constraints above.

We use the following reward structure, which is dependent on the transition:

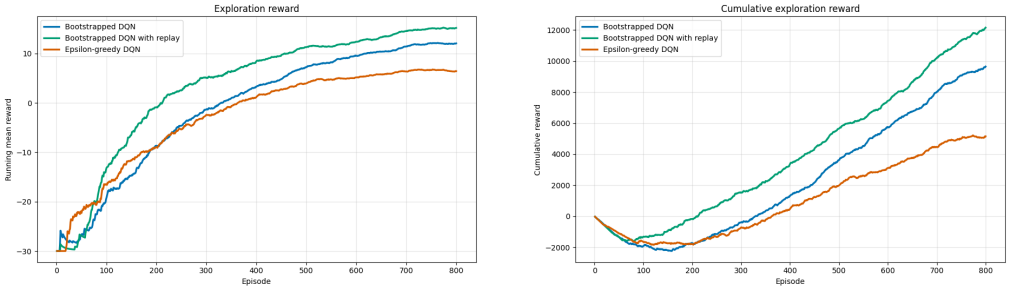
$$R(s, a, s') = \begin{cases} 10, & \text{if } s' = 10, \\ -1, & \text{otherwise.} \end{cases}$$

Therefore, the agent is rewarded for reaching and maintaining the ideal post-demand inventory level of 10, which is the target. The environment is continuing and contains no terminal states, while each exploration episode has 30 steps. The hyperparameters for the inventory optimisation experiment is given in Table 4.6.

For this experiment, we additionally evaluate the proposed Bootstrapped DQN exploration method without the variant of replay buffer $\mathcal{B}$. Figure 4.7 shows the diverse exploration methods. Even though we observe the competitive nature of all exploration methods, we notice that the reply buffer performs slightly better in both mean reward and cumulative reward. However, this does not directly ensure the collection of more informative transition data at this stage, which will be further examined at the next ambiguity set construction.

Table 4.6: Hyperparameters used for inventory optimisation experiment

Parameter	**Value**
Train iterations/episodes	800
Steps per iteration/episode	30
Bootstrapped heads K	10
Episodes to update model τ	100
Mask bootstrapped for Bernoulli p	0.5
Minibatch size b ($\|\mathcal{M}\| = \text{b}$)	64
Non-informative prior a_i	1
Discount factor γ	0.99
Learning rate	0.001
Hidden layer	(32)
Optimiser	Adam
Activation	ReLU
Confidence level for Bayesian	95%
ϵ-greedy Q-learning	0.2



(a) (b)

Figure 4.7: Exploration for Inventory problem: (a) Mean reward for each exploration method and (b) Cumulative reward for each exploration method

Table 4.7: Inventory problem: comparison of robust planning methods for 800 training episodes over 10 independent runs. Entries report the mean ± sample standard deviation. Bold entries identify the best learned method for the corresponding column. As mentioned: $A = \min_{P \in \mathcal{P}} V_P^{\widehat{\pi}}(s_0)$, $B = V_{P^\star}^{\widehat{\pi}}(s_0)$, and $C = V_{P^\star}^{\pi^\star}(s_0)$.

Method	**Robust certificate** A	**True-policy value** B	**Robust-return error** $\lvert C - A \rvert$	**Certificate gap** $\lvert B - A \rvert$	**Policy-oracle gap** $\lvert C - B \rvert$
Bootstrapped DQN with global Bayesian ψ	3.73 ± 5.64	116.11 ± 8.63	154.71 ± 5.64	112.38 ± 4.45	42.33 ± 8.63
Bootstrapped DQN with replay buffer and global Bayesian ψ	$\mathbf{8.58 \pm 15.69}$	$\mathbf{121.17 \pm 20.93}$	149.86 ± 15.69	112.59 ± 7.72	37.27 ± 20.93
Bootstrapped DQN with Hoeffding ψ	-25.85 ± 5.57	117.33 ± 10.71	184.30 ± 5.57	143.18 ± 10.89	41.11 ± 10.71
DQN ϵ-greedy with global Bayesian ψ	-4.79 ± 14.20	118.59 ± 16.16	163.24 ± 14.20	123.38 ± 6.13	39.85 ± 16.16
DQN ϵ-greedy with Hoeffding ψ	-31.83 ± 10.66	120.83 ± 17.61	190.27 ± 10.66	152.66 ± 9.09	37.61 ± 17.61
Simulator benchmark: $C = V^{\pi^\star}(P^\star, s_0) = 158.44 \pm 0.00$					

Due to the highly conservative outcome of previous classical methods, we decided to keep and evaluate only the top performing methods for this experiment. Each method was evaluated over 10 independent runs, mainly due to the high computational time of this more complex experimental setting, of 800 episodes and 30 steps per episode, corresponding to 24,000 environment interactions per run. As a result, we present the results in Table 4.7. In addition, Figure 4.8 visualizes the above results, specifically highlighting the evolution of mean worst case return A when the number of episodes increases, its total robust-return error for each exploration checkpoint, but also on final metrics. The proposed replay-based method achieved the highest mean robust certificate A and true policy B, while remaining competitive under the other evaluation metrics. However, we have to highlight that the overlapping confidence intervals do not support a claim of uniform superiority for this experiment.

Moreover, the learned policies are presented in Figure 4.9, with a specific single seed in Figure 4.9a and across seed fraction of the 10 independent seeds for evaluating the consistency of possible actions in Figure 4.9b. In general, our approach demonstrated more consistent actions and agreement throughout the 10 independent runs in contrast to ϵ-greedy alternatives, with more noticeable disagreement in states $s_{10} - s_{15}$.

Lastly, we evaluate the final learned policies for the best methods under stress scenarios, and specifically external demand conditions. This implies that we did not retrain the agents from scratch or reconstruct their ambiguity sets. Thus, the policies obtained after the initial 800 exploration episodes are held fixed and evaluated

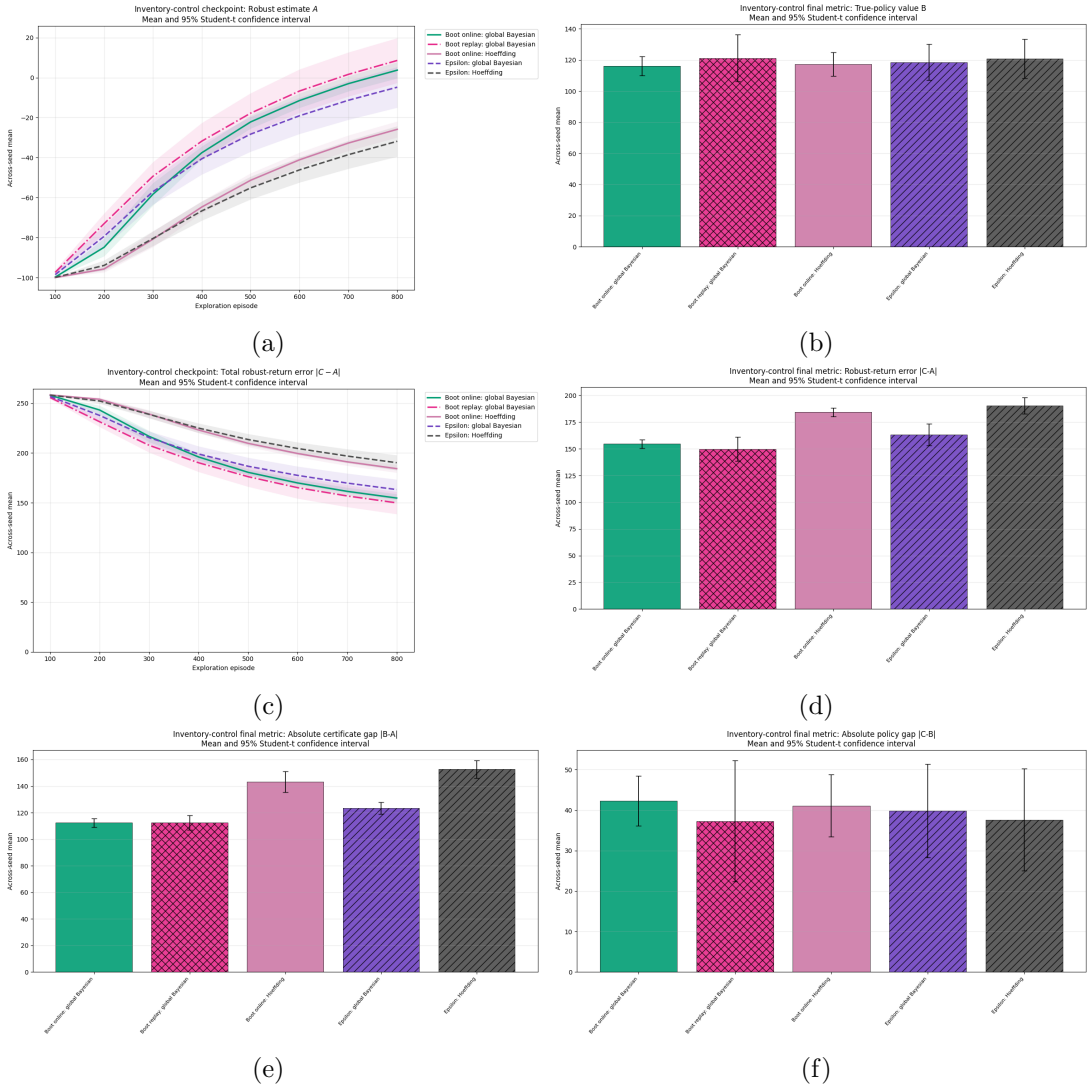
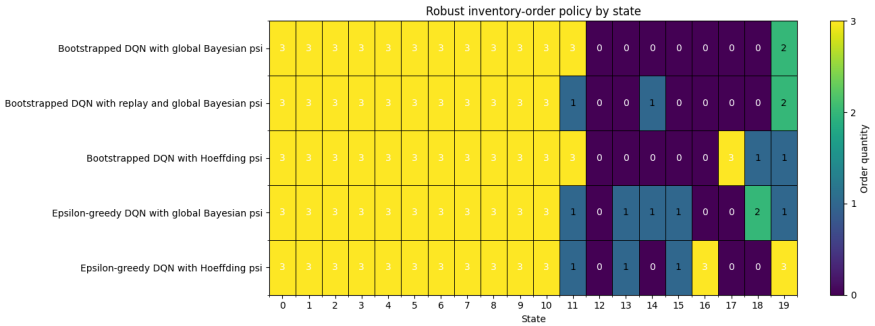


Figure 4.8: Inventory optimisation: results over 10 independent runs, showing means and 95% Student-t confidence intervals as shaded regions in (a) and (c), while error bars in (b),(d)–(f). Specifically: (a) the robust certificate A, (b) the final true-model value B of the learned robust policy, (c) its evolution of error as additional exploration data are collected, (d) its final value for each method. Lower values indicate that the robust certificate is closer to the oracle return, (e) the absolute certificate gap $|B - A|$, and (f) the absolute policy-oracle gap $|C - B|$. Higher values are preferable in (a), while lower values are preferable in (b)-(f).



(a)



(b)

Figure 4.9: Policy schedule for Inventory optimisation after 800 training episodes: (a) policy for a single seed and (b) policy consistency with percentage of success for taking consistent actions, across 10 seeds.

using transition kernels generated from an independent nominal sample. Specifically, the stress scenarios involve changing accordingly the demand mean and variance from equation 4.13. Also, the demand shock scenarios add four units to randomly selected demand observations for 10% and 20% of observations respectively. Table 4.8 reports the mean return and sample standard deviation across 10 seeds for $V^{\widehat{\pi}}_{P_{stress}}(s_0)$. Generally, the proposed method achieves the highest mean return in some stress scenarios and remains competitive in others. However, once again, the overlapping variability does not support a claim of uniform superiority. Instead, the results demonstrate that performance is dependent on the scenario. Increasing the initial number of exploration episodes, instead of 800, when training may reduce the variability and improve policy,

although this is not guaranteed and would imply additional computational cost for each run.

Table 4.8: Inventory problem: stress-test evaluation of policies learned after 800 training episodes over 10 independent runs. Entries report the mean ± sample standard deviation. Bold entries identify the largest mean return among the learned methods for each scenario. The oracle policy under $P^\star$ is reported as a benchmark. Stress: $V^{\widehat{\pi}}_{P_{stress}}(s_0)$

Method	**Stress** $N(2,1)$	**Stress** $N(2,4)$	**Stress** $N(3,2)$	**Stress: 10% demand shocks (+4 units)**	**Stress: 20% demand shocks (+4 units)**
Bootstrapped DQN with global Bayesian ψ	170.12 ± 28.91	58.17 ± 0.85	$\mathbf{-55.17 \pm 8.22}$	45.46 ± 4.10	-19.87 ± 3.93
Bootstrapped DQN with replay buffer and global Bayesian ψ	$\mathbf{188.80 \pm 38.12}$	57.45 ± 11.12	-57.15 ± 6.57	$\mathbf{49.11 \pm 6.92}$	$\mathbf{-18.06 \pm 3.15}$
Bootstrapped DQN with Hoeffding ψ	171.10 ± 30.14	58.96 ± 3.06	-55.18 ± 8.22	45.83 ± 4.90	-19.82 ± 3.98
DQN ϵ-greedy with global Bayesian ψ	182.27 ± 35.86	58.16 ± 3.26	-59.07 ± 12.65	45.26 ± 13.54	-21.11 ± 9.00
DQN ϵ-greedy with Hoeffding ψ	183.59 ± 36.70	$\mathbf{60.16 \pm 4.60}$	-59.07 ± 12.65	45.88 ± 13.92	-21.02 ± 9.05
Oracle policy under $P^\star$	270.94 ± 0.00	68.78 ± 0.00	-62.59 ± 0.00	59.00 ± 0.00	-17.13 ± 0.00

Overall, the inventory experiment, which involves a larger stochastic environment, demonstrates that replay-based Bootstrapped DQN combined with globally calibrated Bayesian ambiguity sets can produce generally less conservative robust certificates, A, while maintaining competitive true-model policy performance, B. The performance of the exported trained policy also performed competitively under stress scenarios. Nevertheless, the variability across runs indicates that the observed advantage is dependent on the domain. Therefore, these results support the competitiveness of the proposed framework rather than universal superiority.

4.5 Concluding remarks

In this chapter, we advanced robust decision-making for Markov Decision Processes whose transition parameters are uncertain because data are limited, noisy or models are misspecified. Most RMDP studies handle either location or size of the ambiguity set, thus producing overly conservative policies. We extended our investigation in fully sequential decision making under uncertainty by, once again, introducing an innovative method to improve the ambiguity set construction in RMDP, which has limited attention. Specifically we provided a data-driven framework that jointly positions and scales the ambiguity set. First, we adapted a Bootstrapped Deep Q-Network to drive deep

exploration and obtained diverse estimates of the transition. The ensemble mean of the bootstrap heads served as an empirically grounded nominal model, which is often a given or fixed set in offline batch settings. Next, from the collected transition counts we derived Bayesian posterior quantiles, thus adapting to uncertainty with larger values where data are noisy and smaller otherwise. The resulting ambiguity sets are compact, showing interesting improvement in performance, mainly in terms of worst-case return, true-model policy performance, comparative error gaps, but also remain tractable. We generated robust policies that dominated classical ambiguity sets and baseline exploration strategies. We are among the first to combine the specific deep exploration method, instead of fixed pre-collected datasets, with Bayesian credible regions to construct ambiguity sets. To the best of our knowledge, the integration of efficient smart exploration before the ambiguity set construction is understudied.

Chapter 5

Conclusions

5.1 Research Findings

In this thesis, we have studied innovative ways to integrate machine learning methods into, and thereby improve uncertainty set construction in robust decision-making under uncertainty, specifically in RO and RMDP. The Big Data revolution and era have driven the creation of systems designed to make decisions from data. After researching the existing literature for the related research areas, we figured out the lack of the innovative use of machine learning models in the uncertainty modelling for both subjects. Particularly, statistical, data-driven and machine learning models have substantially advanced our ability to perform prediction and estimation in those areas, thus it inevitably drew the attention of promising studies, however, they still remain underexplored. The central challenge is how to properly manage the robustness-optimality trade-off, due to the fact that safeguarding against worst-case performance can produce conservatism, especially by traditional state-of-art ideas. On the other hand, we should also aim to ensure feasibility guarantees, but also produce superior performances in both RO and RMDP settings.

In general, both key research areas fall under the broader umbrella of robust decision-making under uncertainty. We wanted to emphasize the complementary investigation of RO and RMDP, and specifically the general applicability of the proposed uncertainty modelling methodology. While these fields can be considered differently, we

did connect and bridged these areas under this data-driven uncertainty set construction frame, which is a vital and natural link. We demonstrated their own importance and their connected idea, while also show how innovative uncertainty modelling approaches can benefit multiple subfields simultaneously. One aspect was the common $\min - \max$ formulation for the optimisation in both fields, specifically in Equation 3.2 for Chapter 3 and in Equation 4.7 for Chapter 4.

Initially, we focused on the role of the uncertainty set construction. We stated that uncertainty plays a crucial role in those fields, including noisy or limited data, modelling errors etc. Then, we examined innovative machine learning models that can be integrated in this vital process. We have discussed other conservative and relevant approaches in the fields, and their implications. Although we did not derive a new theory from scratch, we did rely on known results and specifically we did cite that our final uncertainty set maintain a certain confidence level. In addition to this, we have provided detailed descriptions of each separate field's problem and how uncertainty modelling influences their solutions. Next, we will present all the research finding for each separate chapter based on their experimental settings.

In Chapter 3, we formally stated the uncertainty set construction within RO and provided a detailed explanation on how data-driven techniques can be incorporated and affect the outcome. Our work advanced decision-making under uncertainty by integrating unsupervised learning clustering with data-driven uncertainty set construction for RO. We then explored two-stage RO, as limited sequential decision-making. This versatile methodology is applicable to various RO problems across multiple domains. We addressed those limitations of traditional methods by employing Dirichlet process mixture model to automatically shape uncertainty sets in RO problems, where each predicted distribution corresponds to a cluster.

Specifically, we decided to proceed with Variational Inference because unlike Gibbs sampling which is an instance of Markov chain Monte Carlo inference, Variational Inference is deterministic. In addition, it is quite easy to gauge convergence because it requires only dozens iterations, instead of thousands of iterations like other methods. We also note that the true posterior distribution required several covariance param-

eters, while the approximated covariance distribution requires far less, while reduces the complexity. We essentially acknowledge that we know we cannot derive the right answer, but we will get the best answer we can under the complexity constraints we can tolerate. The estimated parameters from the approximated surrogate posterior were used in the union of several uncertainty set constructions.

Furthermore, this the first time we have shown that the integration and introduction of bootstrap resampling within this process significantly improves the solution quality of our solutions. The results of our examples demonstrate the effectiveness of our approach, outperforming existing methods for the specific cases in terms of objective value at the ideal coverage rates. Consequently, this chapter merges machine learning and optimisation by employing the bootstrapped Dirichlet Process Mixture Models to build uncertainty sets for RO, directly integrating predictive machine learning with prescriptive optimisation.

Following this, in Chapter 4 we extended to sequential Decision-Making and we explored how proper statistical methods can improve the ambiguity set construction in RMDP. Particularly, we introduced an RMDP framework which is designed to address the main challenge of conservative policies. To tackle this, we constructed compact ambiguity sets that better handle the actual uncertainty presented and properly optimise both the size and location of ambiguity sets. We integrated an ensemble Q-learning algorithm, Bootstrapped DQN, to generate a robust set of trajectories and thus collect more representative transition probabilities. We then employed Bayesian models to construct tight and compact ambiguity sets using those pre-collected data.

Moreover, this is the first work to combine the specific deep exploration method, instead of pre-collected datasets, in an online setting with Bayesian credible regions to construct ambiguity sets for RMDP. What we add beyond prior works is that unlike purely offline RMDP settings, where dataset $\mathcal{D}$ is fixed or given, we assume generative access to the environment and thus study how exploration affects the data-driven ambiguity set construction. For instance, in Petrik and Russel (2019); Petrik and Subramanian (2016) the authors assumed that there is a given dataset $\mathcal{D}$. This is quite convenient but also challenging to accurately collect in real life scenarios. In that set-

ting, the only requirement is how we build a data-driven ambiguity set $\mathcal{P}$ from $\mathcal{D}$. We are addressing a slightly different setting: we still assume samples come from the true but unknown P^*, but we choose how to collect $\mathcal{D}$ in an online deep exploration within a simulator, essentially employing an online RRL in the RMDP framework. This extra step is vital because the main elements of an ambiguity set, centre and radius, depend on the data. Essentially, we used deep exploration and Bayesian modelling to improve the RMDP uncertainty estimation under diverse environments, while presenting several evaluations, comparisons, and stress scenarios. Consequently, we can argue that only a few works have constructed ambiguity sets in this specific way and no prior work has integrated deep exploration into Bayesian RMDPs in this manner.

To sum up, our findings for both fields show promising improvement in the uncertainty set construction. The proposed uncertainty and ambiguity sets are compact in the standard mathematical sense. Separately, when we describe them as "more compact" than alternative constructions, we refer to their tight structure and well-defined geometry, which exclude unnecessary regions. We specifically let the data-driven methods guide this process and we essentially designed them to be as small as possible while still capturing all plausible uncertainty realisation. This is in contrast to conventional and overly conservative uncertainty sets which omit this. Furthermore, as discussed, we wanted to ensure that they remain theoretically supported with the relevant probabilistic guarantees in RO and safely estimating the policy return in RMDP.

Lastly, we map the **Research Objectives** stated in *Chapter 1* to the main outcomes established in *Chapters 3* and *Chapter 4*. The mapping clarifies how each objective is met and how the resulting contributions collectively advance data-driven uncertainty modelling for robust decision-making:

- **RO.1** is met in *Chapter 3*, where a data-driven uncertainty set construction is developed within RO models. The results showed that the resulting data-driven uncertainty sets remain compact, while they still produced improved decisions relative to conventional uncertainty set in terms of objective value. This supports the claim that the proper design of uncertainty sets is central to balancing conservatism and performance (see Figure 3.7, Figure 3.8).

- *Chapter 3* satisfies the **RO.2** by employing nonparametric Bayesian mixture modelling, DPMM, to capture complex distributional features in the observed data. Integrating this learned structure into the uncertainty set led to superior results, specifically data-driven uncertainty sets that reflect the data better than the baselines (see Algorithm 1, Figure 3.1).

- **RO.3** is met in *Chapter 3* because the introduction of bootstrap resampling offered a great robustness by reducing the sensitivity of the estimated parameters to variation. The objective was slightly different but the advantage of robustness is justified, leading to upcoming reliable RO solutions (see Algorithm 2, Figure 3.5, Figure 3.12).

- *Chapter 4* meets the **RO.4** by constructing ambiguity sets for transition uncertainty using Bayesian credible regions. The specific ambiguity set construction draws a lot of attention in the field, while it still supports tractable robust planning with l_1 norm. The results include robust value functions and robust policies (see Algorithm 5).

- **RO.5** is addressed in *Chapter 4* by integrating Bootstrapped DQN as an online exploration strategy to the RMDP in order to generate more informative transition trajectories. Bootstrapped DQN exploration shaped the distribution of collected state–action visits, which in turn influenced the nominal transition estimates (centre) and the Bayesian uncertainty (radius) (see Algorithm 4, Table 4.2, Figure 4.1).

- **RO.6** is satisfied by the computational experiments across *Chapter 3* and *Chapter 4*. In RO settings, the proposed compact uncertainty set produced improved objective values under uncertainty compared to conservative uncertainty set constructions (see Table 3.1, Table 3.3). In sequential settings, the proposed data-driven ambiguity set framework improved the overall worst-case return while reduced unnecessary conservatism in contrast to traditional RMDP baselines (see Table 4.3, Figure 4.4)

5.2 Research Limitations

Like several other studies, it is inevitable not to have any limitation during this research period that we need to highlight. Providing research limitations is just as essential as presenting the above findings, as they offer important insights and contribute to the overall conclusions of the study. The major limitation we faced in this dissertation is centred around trying to find a good balance between conservativeness and performance for both fields.

Specifically, for the uncertainty set construction in RO, the effectiveness of the approach heavily relies on the quality and availability of data. In scenarios with limited data, our constructed uncertainty sets still accurately captured the true underlying uncertainties that resulted in a minor improvement in the objective value, but at the cost of increased computation time. Particularly, the posterior inference for DPMMs is computationally more demanding than simpler clustering methods such as k-means or DBSCAN. The introduction of bootstrapped resampling in DPMM led to interesting and innovative results, but at some cases, like in Figure 3.12, the objective value of non-bootstrap approach was slightly higher than the bootstrap approach. However, the small difference and the robustness of our approach make it a better overall choice for generalisation, while it makes our approach robust enough to cover any potential data or parameter perturbations that other methods might miss. This highlights the difficulty researchers face in balancing and finding a trade-off between conservativeness and optimality. Also, while Variational Inference performed exceptional, it is a biased estimate of the true posterior. Moreover, in cases where our model predicted a large number of clusters, computational costs increased substantially, while the performance could degrade on high-dimensional problems and thereby became quite challenging to solve. It would also be quite interesting to compare with even more recent uncertainty set construction methods rather than modifications of classical uncertainty sets. Specifically, the clustering approaches employed for comparison, like k-means and DBSCAN, even though they are widely used, they do have their own limitations. For instance, k-means is more suitable for simpler geometric clustering baseline and requires the

number of clusters to be determined in advanced, while DBSCAN was sensitive to its hyperparameters due to its density nature. Thus a stronger probabilistic baselines, that might account the uncertainty, can be employed in a future work.

Similarly, we encounter similar limitations and issues in the ambiguity set construction in RMDP. Our experimental results demonstrated promising improvements in contrast to classical approaches, however, the resulting policies still remained conservative. This is acknowledged as a general limitation in the field by several authors, like Chen and Li (2020); Derman et al. (2020); Petrik and Russel (2019); Petrik and Subramanian (2016). On top of this, it is a well known that rectangular ambiguity sets, while they remain tractable and standard approaches, they are still overall conservative due to this (s, a)-rectangular assumption which chooses worst-case transition independently at each state-action pair. Also, our work focused on the uncertainty in transition probabilities, and we did not explore at all any reward uncertainty or other extension. Furthermore, our framework assumes the ability to interact with the environment in order to collect trajectories, while in offline settings with a fixed logged dataset, the exploration component cannot improve the data. Once again, this comes with the high computational cost due to the complexity involved, while as the number of K increases, runtime grows substantially, and the method becomes expensive in larger environments. In addition, our results showed a noticeable sensitivity to different hyperparameters, such as the number of heads K, that results in a high variance and CI in Table 4.5. Even though this is typical in DRL (Henderson et al., 2018; Zouitine et al., 2024) and we do have some concrete results to rely on, the outcome varied across random seeds and the method still shows some unstable behaviour in some runs. Lastly, our evaluation is performed on benchmark simulated environments. While this is a standard approach for RMDP research, the absence of real-world case studies limits conclusions about its practicality.

Ultimately, we have to highlight that there is no universal uncertainty or ambiguity set that perfectly addresses all scenarios and decision-making contexts. Specifically, the complexity of real-world problems demand the adjustment of the uncertainty set construction to particular problem structures and modelling assumptions. Nevertheless,

the data-driven methodologies developed in this thesis provide both innovative and flexible frameworks to construct uncertainty and ambiguity sets directly informed by available data and empirical observations. We acknowledge that no set can universally achieve optimal coverage and there is no "one-size-fits-all" solution that we can rely, however, the data-driven and statistical approaches introduced in this thesis offer a robust and adaptive solution that consistently performs well across various applications and domains. We achieved this by explicitly allowing observed data to guide this process, making our methods data-driven and shape the uncertainty structure for both fields under the holistic idea of decision-making under uncertainty, even though this comes with a significant computational cost.

5.3 Future Research

Finally, here in this section, we are going to discuss the potential future work which may use this research as guidance. Most of the suggestions are related to the key element of the proper uncertainty set construction in each field that accounts the uncertainty involved.

For RO, future research can expand on our methodology to build RO models with automated training and validation procedures. We also decided to not investigate deeper the exported parameter π_k from the approximated surrogate distribution, apart from the selection of the clusters. It would be quite interesting to integrate the specific parameter in the constraints of the problems, following similar ideas of weighted uncertainty sets like mentioned in the literature. With that there is also the prospect of exploring weight allocation on data points, apart from clusters, through weighted stick-breaking modifications or weighted kernels. Furthermore, handling data sparsity more effectively is an area for further investigation, while a future research can explore methods for an appropriate and ideal coverage rate to set in advance. Also, although the reported coverage rates, cross-validation and unseen-data experiments provide an empirical evidence that the constructed uncertainty sets capture the relevant observed uncertainty structure, a comparison of out-of-sample objective value, worst-case performance or multiple coverage levels are left for future research. Additionally, in order to

approximate the posterior distribution of the parameters, other optimisation methods can be evaluated, apart from Variational Inference, which should remain computationally tractable. There is also the prospect to research the most appropriate technique for expanding the boundary of the uncertainty set in order to account for all vital data variations, which is something we successfully investigated by introducing bootstrapped resampling. Moreover, we can evaluate the performance of our framework on even larger and high-dimensional experiments, which would be challenging to solve, but would also help us understand more the prospect and thus forming new methodologies to potentially mitigate this. Lastly, advanced deep learning methods can be employed in RO to calculate more optimal coefficients and provide valuable insights. As mentioned, an active research is finding the right balance between conservativeness and performance, which is quite challenging. Consequently, those coefficients are very important and influence this trade-off, thus new methodologies can be employed towards this direction, while remain theoretically sound.

On the other hand, for RMDP, future works can also attempt to find a better balance between performance and robustness and, in general, less conservative safe policies. As we discussed in the literature, rectangularity is a known tractability restriction and a known source of conservatism. Our framework uses rectangularity to keep robust Bellman planning tractable, and thus a natural extension would be to investigate less conservative ambiguity sets and extend beyond (s,a)-rectangular sets. Furthermore, in this work, we have investigated online RRL for RMDP, thus it is inevitable not to research other approximation methods or diverse exploration techniques to potentially integrate into the centre of the ambiguity set. In addition, further works can integrate reward uncertainty and also re-evaluate the l_1 norm for the ambiguity set construction. Evaluating models over a larger number of random seeds/runs is also highly recommended, although this inherently increases total computation time. Moreover, since our focus was on simulation, we can perform further research on real-life experiments, even though this would result in even higher computational cost. Finally, our framework can also be evaluated on even larger-scale domains, where S is large, and thus richer benchmarks.

Bibliography

Akartunalı, K. and Dauzère-Pérès, S. (2022), 'Dynamic lot sizing with stochastic demand timing', *European Journal of Operational Research* **302**(1), 221–229.

Arabsheybani, A., Arshadi Khamseh, A. and Pishvaee, M. S. (2024), 'Sustainable cold supply chain design for livestock and perishable products using data-driven robust optimization', *International Journal of Management Science and Engineering Management* pp. 1–16.

Attila, Ö. N., Agra, A., Akartunalı, K. and Arulselvan, A. (2021), 'Robust formulations for economic lot-sizing problem with remanufacturing', *European Journal of Operational Research* **288**(2), 496–510.

Auer, P., Jaksch, T. and Ortner, R. (2008), 'Near-optimal regret bounds for reinforcement learning', *Advances in neural information processing systems* **21**.

Baird, G. (2020), 'Optimising darts strategy using markov decision processes and reinforcement learning', *Journal of the Operational Research Society* **71**(6), 1020–1037.

Ben-Tal, A., El Ghaoui, L. and Nemirovski, A. (2009), *Robust Optimization*, Vol. 28, Princeton university press.

Ben-Tal, A., Goryashko, A., Guslitzer, E. and Nemirovski, A. (2004), 'Adjustable robust solutions of uncertain linear programs', *Mathematical programming* **99**(2), 351–376.

Ben-Tal, A. and Nemirovski, A. (1998), 'Robust convex optimization', *Mathematics of operations research* **23**(4), 769–805.

Ben-Tal, A. and Nemirovski, A. (1999), 'Robust solutions of uncertain linear programs', *Operations research letters* **25**(1), 1–13.

Ben-Tal, A. and Nemirovski, A. (2000), 'Robust solutions of linear programming problems contaminated with uncertain data', *Mathematical Programming* **88**, 411–424.

Ben-Tal, A. and Nemirovski, A. (2002), 'Robust optimization–methodology and applications', *Mathematical Programming* **92**, 453–480.

Bertsimas, D., Brown, D. B. and Caramanis, C. (2011), 'Theory and applications of robust optimization', *SIAM Review* **53**(3), 464–501.

Bertsimas, D. and de Ruiter, F. J. (2016), 'Duality in two-stage adaptive linear optimization: Faster computation and stronger bounds', *INFORMS Journal on Computing* **28**(3), 500–511.

Bertsimas, D., Den Hertog, D. and Pauphilet, J. (2021), 'Probabilistic guarantees in robust optimization', *SIAM Journal on Optimization* **31**(4), 2893–2920.

Bertsimas, D. and Georghiou, A. (2015), 'Design of near optimal decision rules in multistage adaptive mixed-integer optimization', *Operations Research* **63**(3), 610–627.

Bertsimas, D., Gupta, V. and Kallus, N. (2018), 'Data-driven robust optimization', *Mathematical Programming* **167**, 235–292.

Bertsimas, D. and Kim, C. W. (2024), 'A machine learning approach to two-stage adaptive robust optimization', *European Journal of Operational Research* .

Bertsimas, D., Shtern, S. and Sturt, B. (2022), 'Two-stage sample robust optimization', *Operations Research* **70**(1), 624–640.

Bertsimas, D. and Sim, M. (2004), 'The price of robustness', *Operations Research* **52**(1), 35–53.

Bertsimas, D. and Thiele, A. (2006), Robust and data-driven optimization: modern decision making under uncertainty, *in* 'Models, methods, and applications for innovative decision making', INFORMS, pp. 95–122.

Bertsimas, D. and Van Parys, B. (2022), 'Bootstrap robust prescriptive analytics', *Mathematical Programming* **195**(1), 39–78.

Birge, J. R. and Louveaux, F. (2011), *Introduction to Stochastic Programming*, Springer Science & Business Media.

Blei, D. M. and Jordan, M. I. (2006), 'Variational inference for dirichlet process mixtures'.

Brown, D., Niekum, S. and Petrik, M. (2020), 'Bayesian robust optimization for imitation learning', *Advances in Neural Information Processing Systems* **33**, 2479–2491.

Burgard, J. P., Moreira Costa, C. and Schmidt, M. (2022), 'Robustification of the k-means clustering problem and tailored decomposition methods: when more conservative means more accurate', *Annals of Operations Research* pp. 1–44.

Campbell, T. and How, J. P. (2015), Bayesian nonparametric set construction for robust optimization, *in* '2015 American Control Conference (ACC)', IEEE, pp. 4216–4221.

Caron, F., Davy, M., Doucet, A., Duflos, E. and Vanheeghe, P. (2007), 'Bayesian inference for linear dynamic models with dirichlet process mixtures', *IEEE Transactions on Signal Processing* **56**(1), 71–84.

Chen, S. and Li, Y. (2020), An overview of robust reinforcement learning, *in* '2020 IEEE International Conference on Networking, Sensing and Control (ICNSC)', IEEE, pp. 1–6.

Chen, Z., Sim, M. and Xiong, P. (2020), 'Robust stochastic optimization made easy with rsome', *Management Science* **66**(8), 3329–3339.

Chu, Y. and You, F. (2013), 'Integrated scheduling and dynamic optimization of complex batch processes with general network structure using a generalized benders de-

composition approach', *Industrial & Engineering Chemistry Research* **52**(23), 7867–7885.

Dai, L., You, D. and Yin, X. (2020), 'Data driven robust energy and reserve dispatch based on a nonparametric dirichlet process gaussian mixture model', *Energies* **13**(18), 4642.

Dai, L., You, D., Yin, X., Wang, G. and Zou, Q. (2019), 'Distributionally robust dynamic economic dispatch model with conditional value at risk recourse function', *International Transactions on Electrical Energy Systems* **29**(4), e2775.

Decker, C. and Sester, J. (2024), 'Robust q-learning for finite ambiguity sets', *arXiv preprint arXiv:2407.04259* .

Derman, E., Mankowitz, D., Mann, T. and Mannor, S. (2020), A bayesian approach to robust reinforcement learning, *in* 'Uncertainty in Artificial Intelligence', PMLR, pp. 648–658.

Derman, E., Men, Y., Geist, M. and Mannor, S. (2023), Robustness and regularization in reinforcement learning, *in* 'NeurIPS 2023 Workshop on Generalization in Planning'.

Dumouchelle, J., Julien, E., Kurtz, J. and Khalil, E. B. (2023), 'Neur2ro: Neural two-stage robust optimization', *arXiv preprint arXiv:2310.04345* .

Faccini, D., Maggioni, F. and Potra, F. A. (2022), 'Robust and distributionally robust optimization models for linear support vector machine', *Computers & Operations Research* **147**, 105930.

Gadot, U., Wang, K., Derman, E., Kumar, N., Levy, K. and Mannor, S. (2023), Targeted uncertainty reduction in robust mdps, *in* 'NeurIPS 2023 Workshop on Generalization in Planning'.

Garcia, D. J. and You, F. (2015), 'Supply chain design and optimization: Challenges and opportunities', *Computers & Chemical Engineering* **81**, 153–170.

Ghavamzadeh, M., Petrik, M. and Chow, Y. (2016), 'Safe policy improvement by minimizing robust baseline regret', *Advances in Neural Information Processing Systems* **29**.

Goerigk, M. and Kurtz, J. (2022), 'Data-driven prediction of relevant scenarios for robust optimization', *arXiv e-prints* pp. arXiv–2203.

Goerigk, M. and Kurtz, J. (2023), 'Data-driven robust optimization using deep neural networks', *Computers & Operations Research* **151**, 106087.

Goerigk, M., Kurtz, J. et al. (2023), 'Data-driven robust optimization using unsupervised deep learning', *Computers & Operations Research* **151**(106087).

Gong, J. and You, F. (2017), 'Optimal processing network design under uncertainty for producing fuels and value-added bioproducts from microalgae: two-stage adaptive robust mixed integer fractional programming model and computationally efficient solution algorithm', *AIChE Journal* **63**(2), 582–600.

Goyal, A. and Grand-Clement, J. (2022), 'Improved algorithms for robust mdps with low-dimensional representations', *arXiv preprint arXiv:2203.16973* .

Hao, Z., He, L., Hu, Z. and Jiang, J. (2020), 'Robust vehicle pre-allocation with uncertain covariates', *Production and Operations Management* **29**(4), 955–972.

Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D. and Meger, D. (2018), Deep reinforcement learning that matters, *in* 'Proceedings of the AAAI conference on artificial intelligence', Vol. 32.

Hwang, U. and Hong, S. (2024), 'On practical robust reinforcement learning: Adjacent uncertainty set and double-agent algorithm', *IEEE Transactions on Neural Networks and Learning Systems* .

Iyengar, G. N. (2005), 'Robust dynamic programming', *Mathematics of Operations Research* **30**(2), 257–280.

Jin, H., Li, Z., Sun, H., Guo, Q., Chen, R. and Wang, B. (2017), 'A robust aggregate model and the two-stage solution method to incorporate energy intensive enterprises in power system unit commitment', *Applied Energy* **206**, 1364–1378.

Jordan, M. I. and Mitchell, T. M. (2015), 'Machine learning: Trends, perspectives, and prospects', *Science* **349**(6245), 255–260.

Julien, E., Postek, K. and Birbil, Ş. İ. (2022), 'Machine learning for k-adaptability in two-stage robust optimization', *arXiv preprint arXiv:2210.11152* .

Kimura, T., Tokuda, T., Nakada, Y., Nokajima, T., Matsumoto, T. and Doucet, A. (2013), 'Expectation-maximization algorithms for inference in dirichlet processes mixture', *Pattern Analysis and Applications* **16**, 55–67.

Li, N., Li, W., Jiang, Y. and Xia, S.-T. (2022), Deep dirichlet process mixture models, *in* 'Uncertainty in Artificial Intelligence', PMLR, pp. 1138–1147.

Lim, S. H., Xu, H. and Mannor, S. (2013), 'Reinforcement learning in robust markov decision processes', *Advances in Neural Information Processing Systems* **26**.

Lobo, E. A., Ghavamzadeh, M. and Petrik, M. (2020), 'Soft-robust algorithms for batch reinforcement learning', *arXiv preprint arXiv:2011.14495* .

Lu, M., Zhong, H., Zhang, T. and Blanchet, J. (2024), 'Distributionally robust reinforcement learning with interactive data collection: Fundamental hardness and near-optimal algorithms', *Advances in Neural Information Processing Systems* **37**, 12528–12580.

Meng, L., Goodwin, M., Yazidi, A. and Engelstad, P. E. (2022), 'Improving the diversity of bootstrapped dqn by replacing priors with noise', *IEEE Transactions on Games* **15**(4), 580–589.

Moos, J., Hansel, K., Abdulsamad, H., Stark, S., Clever, D. and Peters, J. (2022), 'Robust reinforcement learning: A review of foundations and recent advances', *Machine Learning and Knowledge Extraction* **4**(1), 276–315.

Neofytou, A., Liu, B. and Akartunalı, K. (2025), 'Compact uncertainty sets for robust optimization based on bootstrapped dirichlet process mixture model', *Optimization* pp. 1–44.

Nilim, A. and El Ghaoui, L. (2005), 'Robust control of markov decision processes with uncertain transition matrices', *Operations Research* **53**(5), 780–798.

Ning, C. and You, F. (2017), 'Data-driven adaptive nested robust optimization: general modeling framework and efficient computational algorithm for decision making under uncertainty', *AIChE Journal* **63**(9), 3790–3817.

Ning, C. and You, F. (2018), 'Data-driven decision making under uncertainty integrating robust optimization with principal component analysis and kernel smoothing methods', *Computers & Chemical Engineering* **112**, 190–210.

Ning, C. and You, F. (2019), 'Data-driven adaptive robust unit commitment under wind power uncertainty: A bayesian nonparametric approach', *IEEE Transactions on Power Systems* **34**(3), 2409–2418.

Osband, I., Blundell, C., Pritzel, A. and Van Roy, B. (2016), 'Deep exploration via bootstrapped dqn', *Advances in neural information processing systems* **29**.

Osband, I., Van Roy, B., Russo, D. J. and Wen, Z. (2019), 'Deep exploration via randomized value functions', *Journal of Machine Learning Research* **20**(124), 1–62.

Owen, A. B. and Eckles, D. (2012), 'Bootstrapping data arrays of arbitrary order'.

Panaganti, K. and Kalathil, D. (2022), Sample complexity of robust reinforcement learning with a generative model, *in* 'International Conference on Artificial Intelligence and Statistics', PMLR, pp. 9582–9602.

Panaganti, K., Xu, Z., Kalathil, D. and Ghavamzadeh, M. (2022), 'Robust reinforcement learning using offline data', *Advances in neural information processing systems* **35**, 32211–32224.

Petrik, M. and Russel, R. H. (2019), 'Beyond confidence regions: Tight bayesian ambiguity sets for robust mdps', *Advances in neural information processing systems* **32**.

Petrik, M. and Subramanian, J. (2016), Safe policy improvement by minimizing robust baseline regret, *in* 'Advances in Neural Information Processing Systems', pp. 2487–2495.

Puterman, M. L. (2014), *Markov decision processes: discrete stochastic dynamic programming*, John Wiley & Sons.

Qin, S. J. (2014), 'Process data analytics in the era of big data', *AIChE Journal* **60**(9), 3092–3100.

Quezada, F., Gicquel, C. and Kedad-Sidhoum, S. (2022), 'Combining polyhedral approaches and stochastic dual dynamic integer programming for solving the uncapacitated lot-sizing problem under uncertainty', *INFORMS Journal on Computing* **34**(2), 1024–1041.

Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S. A., Binder, A., Müller, E. and Kloft, M. (2018), Deep one-class classification, *in* 'International Conference on Machine Learning', PMLR, pp. 4393–4402.

Russel, R. H., Behzadian, B. and Petrik, M. (2019), 'Optimizing norm-bounded weighted ambiguity sets for robust mdps', *arXiv preprint arXiv:1912.02696* .

Sahinidis, N. V. (2004), 'Optimization under uncertainty: state-of-the-art and opportunities', *Computers & Chemical engineering* **28**(6-7), 971–983.

San Juan, J. L. and Sy, C. (2024), 'A data-driven target-oriented robust optimization framework: bridging machine learning and optimization under uncertainty', *Journal of Industrial and Production Engineering* pp. 1–25.

Satia, J. and Lave, R. (1973), 'Markovian decision processes with uncertain transition probabilities', *Operations Research* **21**(3), 728–740.

Shang, C., Huang, X. and You, F. (2017), 'Data-driven robust optimization based on kernel learning', *Computers & Chemical Engineering* **106**, 464–479.

Shang, C. and You, F. (2019), 'A data-driven robust optimization approach to scenario-based stochastic model predictive control', *Journal of Process Control* **75**, 24–39.

Soyster, A. L. (1973), 'Convex programming with set-inclusive constraints and applications to inexact linear programming', *Operations Research* **21**(5), 1154–1157.

Strehl, A. L. and Littman, M. L. (2008), 'An analysis of model-based interval estimation for markov decision processes', *Journal of Computer and System Sciences* **74**(8), 1309–1331.

Sutton, R. S. (2018), 'Reinforcement learning: An introduction', *A Bradford Book* .

Tay, D. H., Ng, D. K. and Tan, R. R. (2013), 'Robust optimization approach for synthesis of integrated biorefineries with supply and demand uncertainties', *Environmental Progress & Sustainable Energy* **32**(2), 384–389.

Thomas, P., Theocharous, G. and Ghavamzadeh, M. (2015), High-confidence off-policy evaluation, *in* 'Proceedings of the AAAI Conference on Artificial Intelligence', Vol. 29.

Thompson, W. R. (1933), 'On the likelihood that one unknown probability exceeds another in view of the evidence of two samples', *Biometrika* **25**(3-4), 285–294.

Tirinzoni, A., Petrik, M., Chen, X. and Ziebart, B. (2018), 'Policy-conditioned uncertainty sets for robust markov decision processes', *Advances in neural information processing systems* **31**.

Uğurlu, K. (2024), 'Terminal wealth maximization under drift uncertainty', *Optimization* pp. 1–19.

Uğurlu, K. and Brzeczek, T. (2023), 'Distorted probability operator for dynamic portfolio optimization in times of socio-economic crisis', *Central European Journal of Operations Research* **31**(4), 1043–1060.

van Hulst, J., Heemels, W. and Antunes, D. (2025), 'Smart exploration in reinforcement learning using bounded uncertainty models', *arXiv preprint arXiv:2504.05978* .

Wang, I., Becker, C., Van Parys, B. and Stellato, B. (2022), 'Mean robust optimization', *arXiv preprint arXiv:2207.10820* .

Wang, S., Si, N., Blanchet, J. and Zhou, Z. (2023), 'On the foundation of distributionally robust reinforcement learning', *arXiv preprint arXiv:2311.09018* .

Weissman, T., Ordentlich, E., Seroussi, G., Verdu, S. and Weinberger, M. J. (2003), 'Inequalities for the l1 deviation of the empirical distribution', *Hewlett-Packard Labs, Tech. Rep* p. 125.

Wiesemann, W., Kuhn, D. and Rustem, B. (2013), 'Robust markov decision processes', *Mathematics of Operations Research* **38**(1), 153–183.

Xu, H. and Mannor, S. (2009), 'Distributionally robust markov decision processes', *Mathematics of Operations Research* **34**(4), 825–844.

Xu, H. and Mannor, S. (2010), Distributionally robust markov decision processes, *in* 'Advances in Neural Information Processing Systems', pp. 2505–2513.

Yao, L., Wang, X., Duan, C., Guo, J., Wu, X. and Zhang, Y. (2018), 'Data-driven distributionally robust reserve and energy scheduling over wasserstein balls', *IET Generation, Transmission & Distribution* **12**(1), 178–189.

Zhou, L., Zhang, L. and Horn, B. K. (2020), 'Deep reinforcement learning-based dynamic scheduling in smart manufacturing', *Procedia Cirp* **93**, 383–388.

Zhu, J., Ge, Z. and Song, Z. (2016), 'Variational bayesian gaussian mixture regression for soft sensing key variables in non-gaussian industrial processes', *IEEE Transactions on Control Systems Technology* **25**(3), 1092–1099.

Zouitine, A., Bertoin, D., Clavier, P., Geist, M. and Rachelson, E. (2024), 'Rrls: Robust reinforcement learning suite', *arXiv preprint arXiv:2406.08406* .