HEC MONTRÉAL
École affiliée à l'Université de Montréal

Conditional Robust Optimization for Data-driven Decision-Making

par
Abhilash Reddy Chenreddy

Thèse présentée en vue de l'obtention du grade de Ph. D. en administration
(spécialisation Sciences de la décision)

Juin 2025

HEC MONTRÉAL
École affiliée à l'Université de Montréal

Cette thèse intitulée :

Conditional Robust Optimization for Data-driven Decision-Making

Présentée par :

Abhilash Reddy Chenreddy

a été évaluée par un jury composé des personnes suivantes :

Okan Arslan
HEC Montréal
Président-rapporteur

Erick Delage
HEC Montréal
Directeur de recherche

Laurent Charlin
HEC Montréal
Membre du jury

Emma Frejinger
Université de Montréal
Membre du jury

Bart Van Parys
Centrum Wiskunde & Informatica
Examinateur externe

Claudia Rebolledo
HEC Montréal
Représentante du directeur de HEC Montréal

Résumé

Une prise de décision fiable dans des environnements incertains nécessite souvent non seulement de la robustesse, mais aussi une adaptabilité au contexte. Ce travail propose une approche unifiée de la *robustesse contextuelle*, en développant des méthodes qui conditionnent la quantification de l'incertitude aux caractéristiques observées, afin d'améliorer la qualité des décisions. Une innovation clé dans cette direction est le développement de l'*Optimisation Robuste Conditionnelle* (Conditional Robust Optimization, CRO), un cadre qui adapte les ensembles d'incertitude à l'information contextuelle. Plutôt que de s'appuyer sur des ensembles d'incertitude statiques représentant les pires cas, cette approche partitionne l'espace des covariables et construit des ensembles d'incertitude spécifiques à chaque région, s'adaptant ainsi à la structure des données. Cette stratégie permet de prendre des décisions qui conservent des garanties de couverture tout en évitant un excès de conservatisme. L'analyse théorique fournit des garanties en termes de risque et de couverture, et les évaluations empiriques démontrent une amélioration des performances par rapport aux méthodes classiques d'optimisation robuste.

Afin d'aligner plus étroitement l'estimation de l'incertitude avec les objectifs de prise de décision, un nouveau cadre d'apprentissage de type *end-to-end* est introduit. Il intègre directement la construction des ensembles d'incertitude à l'objectif d'optimisation en aval. En apprenant conjointement la couverture conditionnelle et la qualité de la décision à l'aide d'une fonction de perte différentiable, cette approche surmonte les problèmes de désalignement inhérents aux méthodes classiques de type *estimer-puis-optimiser*. Le résultat est une méthode flexible et guidée par les données, qui améliore à la fois la robustesse et la performance de la tâche, comme le démontrent les résultats empiriques.

Ce cadre est ensuite étendu à la prise de décision séquentielle, où l'incertitude doit être quantifiée non plus sur des résultats immédiats, mais sur des estimations de valeurs à long terme. Cela mène naturellement au cadre de l'*apprentissage par renforcement hors ligne*, où un agent doit apprendre des politiques à partir de jeux de données statiques, sans interaction supplémentaire avec l'environnement. Pour répondre aux défis uniques de ce contexte, nous développons un cadre général de *robustesse épistémique contextuelle*, qui remplace l'estimation conventionnelle de l'incertitude par des ensembles d'incertitude structurés, conditionnés à l'état, sur les valeurs-Q. Cette formulation généralise les principes de robustesse conditionnelle aux environnements dynamiques, afin de permettre l'apprentissage de politiques efficaces à partir de données hors ligne. Les résultats empiriques sur une variété de tâches démontrent que cette méthode améliore la robustesse et les performances hors distribution, comparativement aux approches basées sur les ensembles.

Ces contributions offrent une perspective unifiée de la robustesse contextuelle, montrant que l'incertitude peut être modélisée comme un objet structuré, apprenable, adaptable aux données et sensible à la tâche décisionnelle. En intégrant des outils issus de l'optimisation robuste, de l'apprentissage statistique et de l'apprentissage par renforcement, ce travail établit une base rigoureuse pour la conception de systèmes décisionnels plus fiables, interprétables et dignes de confiance dans des environnements incertains.

Mots-clés

Optimisation robuste, Optimisation contextuelle, Quantification de l'incertitude, Prédiction conforme, Apprentissage orienté tâches, Apprentissage par renforcement hors ligne.

Méthodes de recherche

Recherche quantitative; Programmation mathématique.

Abstract

Reliable decision making in uncertain environments often requires not just robustness, but also adaptability to context. This work develops a unified approach to contextual robustness, proposing methods that condition uncertainty quantification on observed features to improve quality of decision making. A key innovation in this direction is the development of Conditional Robust Optimization (CRO), a framework that adapts uncertainty sets to contextual information. Instead of relying on static, worst-case uncertainty sets, the approach partitions the covariate space and constructs region specific uncertainty sets that adapt to the structure of the data. This leads to decisions that maintain coverage guarantees while avoiding excessive conservatism. Theoretical analysis provides guarantees on risk and coverage, and empirical evaluations demonstrate improved performance over classical robust optimization methods.

To further align uncertainty estimation with decision making objectives, a new End-To-End learning framework is introduced that directly integrates uncertainty set construction with the downstream optimization objective. By jointly learning conditional coverage and decision quality using a differentiable surrogate loss, this approach overcomes the misalignment issues inherent to the traditional Estimate-Then-Optimize methods. The result is a flexible, data driven approach that consistently improves both robustness and task performance, as shown in experimental results.

We further extend this framework to sequential decision making, where uncertainty must be quantified over long term value estimates rather than immediate outcomes. This naturally leads to the setting of offline reinforcement learning, in which an agent must learn policies from static datasets without additional environment interaction. To address the unique challenges of this setting, we develop a general framework for contextual epistemic robustness that replaces conventional ensemble based uncertainty estimation with structured, state conditional uncertainty sets over Q-values. This formulation generalizes the principles of conditional robustness to dynamic environments to learn efficient policies from offline data. Empirical results across a range of tasks demonstrate that this method offers improved robustness and out-of-distribution performance compared to ensemble-based baselines.

These contributions present a unified perspective on context aware robustness showing that uncertainty can be modeled as a structured, learnable object adaptable to data and responsive to the decision making task. By integrating tools from robust optimization, statistical learning, and reinforcement learning, this work establishes a principled foundation for building more reliable, interpretable, and trustworthy decision systems in uncertain environments.

Keywords

Robust Optimization, Contextual Optimization, Uncertainty Quantification, Conformal Prediction, Task based Learning, Offline Reinforcement Learning.

Research Methods

Quantitative research; Mathematical programming.

Contents

List of Tables

List of Figures

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my advisor, Prof. Erick Delage. His unwavering support, intellectual rigor, and generosity with his time have shaped my journey in countless ways. More than just a supervisor, he has been a model for how to lead a meaningful and impactful research career, an example I will always aspire to follow.

I am sincerely thankful to the members of my thesis committee, Prof. Emma Frejinger and Prof. Laurent Charlin, for their time, thoughtful feedback, and invaluable insights that helped strengthen this work.

I owe special thanks to Prof. Selva Nadarajah, my master's advisor, whose early mentorship and encouragement were instrumental in steering me toward a research career.

I am fortunate to have been part of the Decision Making under Uncertainty research group, whose stimulating discussions and collaborative spirit made research both productive and enjoyable. I am equally grateful to the broader GERAD community, whose academic vibrancy and openness created an inspiring space to grow as a researcher.

To my friends, both those in India and those I have met during my time in the U.S. and Canada, thank you for your friendship, constant encouragement, and the sense of stability you brought to this journey. A special thanks to my oldest friend, Durvas, whose own doctoral journey has been a constant source of inspiration and motivation.

To my partner and best friend, Nymisha, thank you for your love, patience, and unwavering support. To my parents, Srinivas and Srilatha, and my sister, Sahithi - I am incredibly lucky to have you in my life. Your constant encouragement, unwavering belief in me, and unconditional support have guided and uplifted me every step of the way. To my family, your love and support have been a steady source of strength throughout this journey, something I will always carry with gratitude. And to my dog, Huckleberry, thank you for your joyful energy and companionship, which brought light to even the longest days.

Finally, I gratefully acknowledge the support of the IVADO Doctoral Excellence Scholarship, which made this research possible.

Preface

This thesis consists of three articles listed as follows:

- Abhilash Reddy Chenreddy, Nymisha Bandi, Erick Delage, Data-Driven Conditional Robust Optimization, Advances in Neural Information Processing Systems, 2022.

- Abhilash Reddy Chenreddy, Erick Delage, End-to-end Conditional Robust Optimization, Proceedings of the Fortieth Conference on Uncertainty in Artificial Intelligence, 2024.

- Abhilash Reddy Chenreddy, Erick Delage, Epistemic Robustness in Offline Reinforcement Learning, to be submitted.

General Introduction

In complex real world environments, businesses are routinely required to navigate uncertainty by making decisions that remain robust in the face of adversity while simultaneously adapting to evolving contextual conditions, thereby ensuring both reliability and responsiveness. Whether in financial planning through portfolio optimization (Xidonas, Steuer, and Hassapis 2020), managing supply chains (Suryawanshi and Dutta 2022; Bertsimas and Thiele 2004) or controlling autonomous systems (Rosolia, Zhang, and Borrelli 2018), presence of uncertainty can result in unreliable or even infeasible solutions. Seminal work by Ben-Tal and Nemirovski (2000) demonstrates that small perturbations in the problem parameters can make the optimal solution infeasible, thereby undermining their practical value. The sources of such uncertainty are varied, ranging from noisy environments, and inconsistencies in data collection processes to the limited availability of high quality data. These challenges become even more pronounced in the modern data-driven, interconnected decision making pipelines where the output of one model often serves as the input to another. These are commonly referred to as Estimate-then-Optimize pipelines, in which predictions from machine learning models are used as inputs in subsequent optimization processes (Elmachtoub et al. 2023). In such settings, even small errors from base models can propagate and compound downstream, raising a fundamental question: *how can we make reliable and trustworthy decisions when the input data to the models can not be fully trusted?*

Stochastic optimization emerged as a natural response to decision making under uncertainty. Foundational works such as Birge and Louveaux (1997) and Shapiro, Dentcheva, and Ruszczynski (2021) present the key formulations together with their analytical properties and associated computational techniques. The central paradigm assumes access to a probability distribution over uncertain parameters and seeks to optimize the expected value of an objective function (see also Li and Grossmann (2021) for a recent survey). When the underlying distribution is well specified, stochastic optimization provides a powerful and elegant framework to identify solutions that perform well on average across possible realizations. The practical effectiveness of stochastic optimization, however, critically depends on the validity of its distributional assumptions. In data driven settings, we are again dependent on the quality and representativeness of the available data used to estimate the distribution. In practice, the empirical distributions used to model uncertainty can misalign with the true environment, especially in cases with limited, noisy, or non-stationary data (Bennouna et al. 2024; Besbes, Gur, and Zeevi 2015). This disconnect can lead to decisions that perform well in expectation under the assumed model but fail systematically when deployed in the real world, making it less robust (Smith and Winkler 2006). Related concerns have motivated alternative paradigms, including distributionally favorable optimization (DFO), which emphasizes optimistic model selection in the presence of endogenous outliers (Jiang and Xie 2023), as well as recent work that highlights the connections between distributionally robust optimization and classical robust statistics (Blanchet et al. 2024). In such cases, the inherent optimism of the expected value optimization becomes a liability rather than an asset, highlighting the need for alternative formulations that are more resilient to data uncertainty.

To address the limitations of distribution based approaches, **Robust Optimization (RO)** offers a compelling alternative that avoids explicit probability models. RO considers a minimax formulation where decisions are optimized for the worst-case realizations of the uncertain parameters within a predefined uncertainty set (Chen, Sim, and Sun 2007). This framework trades off average case performance for worst case guarantees, making it suitable for deployment in poorly understood real world environments. In RO, uncertainty is modeled using a deterministic, typically compact and convex uncertainty set that encompasses all plausible realizations of uncertain parameters. The objective is to identify solutions that minimize the maximum possible loss over this set. When both the objective and the uncertainty set are convex, the minimax problems admit tractable reformulations via duality, enabling scalable solutions with linear, second-order cone, or semidefinite programming (Beck and Ben-Tal 2009).

Robust Optimization has found widespread application across various domains, where uncertainty sets are commonly constructed using budgeted uncertainty sets (Bertsimas and Sim 2003), ellipsoidal approximations (Ben-Tal et al. 2011), or confidence regions derived from statistical estimations (Bertsimas, Gupta, and Kallus 2018; Goerigk and Kurtz 2020). However, classical formulations often rely on fixed, global uncertainty sets that are applied uniformly across all problem instances. This design choice can lead to overly conservative decisions, as it ignores contextual or structural dependencies among parameters that may vary with observable features or data specific characteristics.

As decision making becomes increasingly data driven, there is growing interest in contextual optimization, a framework that adapts not only to uncertainty but also to the specific context or features observed at decision time (Sadana et al. 2025; Mandi et al. 2024). In many practical settings, such as personalized pricing, healthcare resource allocation, or demand forecasting, uncertainty is not uniform across instances but varies with covariates like location, demographic profiles, or time. Ignoring such structure leads to overly generic and potentially suboptimal decisions. This resulted in growing interest in research into context aware optimization frameworks that incorporate observable features into both predictive and prescriptive models. Similar ideas have emerged in fields like contextual bandits, meta-learning (Zhou 2015; Lemke, Budka, and Gabrys 2015) where both decision policies and uncertainty estimates are adapted to instance-specific features.

In Chapter 1, we propose **Conditional Robust Optimization (CRO)** as a novel contribution that unifies ideas from contextual optimization and robust decision-making.This work was published in Advances in Neural Information Processing Systems (NeurIPS 2022) (Chenreddy, Bandi, and Delage 2022). CRO generalizes classical robust optimization by allowing uncertainty sets to be conditioned on observed features enabling adaptive and data-driven robustness. Rather than optimizing against a single worst-case distribution, CRO seeks decisions that are robust to context-dependent uncertainty (Chenreddy, Bandi, and Delage 2022). Our approach provides the framework to learn region specific uncertainty sets from data while maintaining statistical coverage guarantees ensuring that the true outcome lies within the set with high probability. We show that, under mild assumptions, the resulting robust optimization problem remains computationally tractable. This contrasts with classical robust optimization, which applies static, global uncertainty sets that often lead to over conservatism. While data driven CRO framework provides a flexible way to incorporate context into uncertainty modeling, it often relies on a two stage pipeline where uncertainty sets are first estimated from data and then used in a separate robust optimization step. While modular and interpretable, this ETO approach often suffers from a misalignment between the loss function used to construct the uncertainty sets and the final decision making objective. A key result also shows that this coverage based construction can be viewed as a contextual extension of value-at-risk optimization, which naturally leads to conservative solutions. This perspective connects to prior observations in the literature (see Lam 2019; Van Parys, Esfahani, and Kuhn 2021).

To overcome this, in Chapter 2, we introduce an end-to-end CRO framework that jointly learns both the uncertainty sets and the optimal decisions. This work is published in the Proceedings of the 40th Conference on Uncertainty in Artificial Intelligence (UAI 2024) (Chenreddy and Delage 2024). By integrating a differentiable optimization layer into the learning process, the model receives direct feedback from the downstream task, allowing it to shape the uncertainty sets in a way that improves decision quality. We also propose a novel joint loss function aimed at enhancing the conditional coverage of the contextual uncertainty sets while optimizing the CRO objective. Empirical results demonstrate that this end-to-end approach surpasses traditional ETO methods in decision quality while achieving comparable or superior conditional coverage.

While the earlier chapters addressed problems aimed at optimizing immediate rewards, these settings assume one time interactions where historical data consist of (ψ, ξ) pairs. The objective is to map covariates ψ to decisions that influence an observed outcome ξ, which directly determines the reward. However, many real-world applications involve sequential decision-making, where actions not only yield immediate returns but also influence future states. In such settings, we may observe tuples of the form (ψ, ξ, ψ', r), where ψ is the current context, ξ the realized outcome conditional on the action, ψ' the subsequent context, and $r \in \mathbb{R}$ the immediate reward. Yet we lack access to the underlying system dynamics or a decision oracle. The goal is to learn a policy that maximizes long-term reward using only logged data. This defines the offline reinforcement learning setting, which differs from online RL in that the agent cannot interact with the environment to collect new data. Offline RL introduces specific challenges, including distributional mismatch between the behavior and target policies, and limited coverage of the state-action space, both of which can lead to poor generalization and unsafe decisions.

In Chapter 3, we introduce Epistemic Robust Soft Actor-Critic (ERSAC), a novel framework that generalizes Conditional Robust Optimization (CRO) to the sequential setting of offline reinforcement learning. In classical CRO, uncertainty sets are conditioned on observable features to induce decisions that are robust to instance-specific variability (Chenreddy, Bandi, and Delage 2022; Y. P. Patel, Rayan, and Tewari 2024). ERSAC extends this principle by constructing state-dependent uncertainty sets over long term expect return values (a.k.a. Q-values), capturing how epistemic uncertainty varies across the state space in dynamic environments.

Traditional approaches to epistemic robustness in offline RL, such as SAC-N, use discrete Q-function ensembles to generate conservative value estimates (An et al. 2021). While effective, these methods rely on sampling-based approximations and require large ensembles to encode directional uncertainty, making them computationally expensive and statistically inefficient.

ERSAC replaces ensembles with a learned, structured uncertainty set $\mathscr{U}(\psi)$ at each state ψ, capturing both the shape and orientation of epistemic variability in Q-values. These sets are parameterized using a scalable variant of Epistemic Neural Networks (Osband et al. 2023), which generate distributions over Q-values conditioned on latent context. This construction enables the learning of robust Bellman backups without relying on bootstrapped ensembles or explicit variance penalization. In doing so, ERSAC preserves the conditional robustness of CRO while introducing a richer and more computationally efficient mechanism for sequential decision-making under uncertainty.

Together, the three chapters of this thesis present a unified perspective on decision making under uncertainty through the lens of data driven and context aware robustness. Chapter 1 introduces a modular framework for learning conditional uncertainty sets from data, enabling robust decisions that adapt to observable covariates. Chapter 2 builds on this foundation by coupling uncertainty estimation with optimization in an end-to-end manner, eliminating the disconnect between prediction and prescription in robust settings. Chapter 3 extends these ideas to the sequential setting, proposing a principled framework for epistemic robustness in offline reinforcement learning, where uncertainty sets are used to stabilize value estimates and improve generalization from static datasets.

Across these settings, we show that treating uncertainty as a structured, learnable object leads to more effective and robust decisions. By bridging robust optimization, statistical learning, and reinforcement learning, this work contributes to a growing body of work that seek to integrate prediction models with decision making frameworks for reliable, interpretable trustworthy decision making in uncertain environments.

Chapter 1

Data-Driven Conditional Robust Optimization

Abstract

In this chapter, we study a novel approach for data-driven decision-making under uncertainty in the presence of contextual information. Specifically, we address this problem using a new Conditional Robust Optimization (CRO) paradigm that seeks the solution of a robust optimization problem where the uncertainty set accounts for the most recent side information provided by a set of covariates. We propose an integrated framework that designs the conditional uncertainty set by jointly learning a partition in the covariate data space and simultaneously constructing region specific deep uncertainty sets for the random vector that perturbs the CRO problem. We also provide theoretical guarantees for the coverage provided by conditional uncertainty sets and for the value-at-risk performances obtained using the proposed CRO model. Finally, we use simulated and real world data to illustrate the implementation of our approach and compare it against two non-contextual robust optimization benchmark approaches to demonstrate the value of exploiting contextual information in robust optimization.

1.1 Introduction

In most real world decision problems, the decision maker (DM) faces uncertainty either in the objective function that he aims to optimize, or some of the constraints that he needs to satisfy. Stochastic Programming and Robust Optimization (RO) are the most popular methods for addressing this issue. With the growing availability of data, there has recently been a surge of interest in modeling optimization under uncertainty as contextual optimization problems that seek to leverage rich feature observations to make better decisions (Ban and Rudin 2019; Bertsimas and Kallus 2020). In a simple cost minimization problem, where $\mathscr{X} \subseteq \mathbb{R}^n$ and $c(x,\xi)$ respectively capture the feasible set of actions and a cost that depends on both the action x and a random perturbation vector $\xi \in \mathbb{R}^m$, the "contextual" DM has access to a vector of covariates $\psi \in \mathbb{R}^m$ assumed to be correlated to ξ. This DM therefore traditionally wishes to identify an optimal policy, i.e. a functional $\boldsymbol{x} : \mathbb{R}^m \to \mathscr{X}$ that suggests an action in $\mathscr{X}$ adapted to the observed realization of ψ, with respect to his expected cost over the joint distribution of (ψ,ξ):

$$\min_{\boldsymbol{x}(\cdot)} \mathbb{E}[c(\boldsymbol{x}(\psi),\xi)]. \tag{1.1}$$

From a theoretical point of view, one can exploit the interchangeability property (see Theorem 14.60 in Rockafellar and Wets (2009)) to identify an optimal policy for Problem (1.1) using the following conditional stochastic optimization (CSO) problem:

$$\text{(CSO)} \qquad \boldsymbol{x}^*(\psi) \in \operatorname*{argmin}_{x\in\mathscr{X}} \mathbb{E}[c(x,\xi)|\psi]. \tag{1.2}$$

While the literature that treats contextual optimization through the CSO problem is rich, much less attention has been given to contextual optimization in the risk averse setting. Namely, one can think about replacing the risk neutral expected value operator in Problem (1.2) with a risk measure such as value-at-risk or conditional value-at-risk in order to prevent the DM from being exposed to the possibility of large costs. Moreover, while robust optimization is being used pervasively in disciplines that employ decision models, including chemical, civil, electrical engineering, medicine, and physics (see respectively Bernardo and Saraiva (1998), Bendsøe, Ben-Tal, and Zowe (1994), Mani, Singh, and

Orshansky (2006), Chu et al. (2005), and Bertsimas, Nohadani, and Teo (2007)) to name a few, the question of how to systematically integrate contextual information in this important class of decision models remains to this day unexplored.

In this work, we therefore tackle for the first time the contextual optimization problem from the point of view of robust optimization. Namely, we will consider a contextual DM that wishes to exploit the side information in the design and solution of a robust optimization problem. This naturally gives rise to the following **conditional robust optimization** (CRO) problem

$$\boldsymbol{x}^*(\psi) := \operatorname*{argmin}_{x \in \mathscr{X}} \max_{\xi \in \mathscr{U}(\psi)} c(x, \xi),$$

where $\mathscr{U}(\psi)$ is an uncertainty set designed to contain with high probability the realization of ξ conditionally on observing ψ. Our proposed approach will be data-driven in the sense that the design of the CRO problem will make use of historical observations of joint realizations of ψ and ξ.

Our contribution can be summarized as follows.

- We propose for the first time a framework for learning from data an uncertainty set for RO that adapts to side information. The “training” of this conditional uncertainty set is done by jointly learning a partition in the covariate data space using deep clustering methods, and simultaneously constructing region specific deep uncertainty sets, using techniques from one-class classification, for the random vector that perturbs the CRO problem.

- We establish theoretical connections between CRO and Contextual Value-at-Risk Optimization (CVO):

  $$\min_{\boldsymbol{x}(\cdot)} \mathrm{VaR}_{1-\varepsilon}(c(\boldsymbol{x}(\psi), \xi)), \tag{1.3}$$

  where $\mathrm{VaR}_{1-\varepsilon}(Z) := \inf\{t | \mathbb{P}(Z \leq t) \geq 1 - \varepsilon\}$ refers to the value-at-risk of $1 - \varepsilon$ confidence level of Z.

- We demonstrate empirically that contextual robust optimization can improve the performance of robust optimization models in a data-driven portfolio optimization problem that employs real world data from the U.S. stock market. In particular, we find that in conditions where side information carries a strong signal about future returns, the risk of the portfolio can be reduced by up to 15%.

The chapter is organized as follows. Section 1.2 surveys related work. Section 1.3 summarizes the approach discussed in Goerigk and Kurtz (2020). Section 1.4 presents a Deep Cluster then Classify (DCC) scheme and our Integrated Deep Cluster then Classify (IDCC) scheme to generate conditional uncertainty sets. It also establishes the connections to CVO. Our case study based on real world portfolio optimization is presented in Section 1.5 followed by conclusions in Section 1.6.

1.2 Related Work

Conditional Stochastic Optimization Hannah, Powell, and Blei (2010) was possibly the earliest work on CSO, where a kernel density estimation approach is exploited to formulate and solve a CSO problem. Ban and Rudin (2019) apply CSO to a newsvendor optimization problem where the performance of linear policies and kernel density estimation is explored and where generalization error can be controlled using regularization. Kallus and Mao (2020) studied methods to train forest decision policies for CSO in a way that directly targets the optimization costs. Ban, Gallien, and Mersereau (2019) use residual tree methods to solve general multi-stage stochastic programs where information about the underlying uncertainty is available through covariate information. Kannan, Bayraksan, and J. R. Luedtke (2020) propose data-driven SAA frameworks for approximating the solution to two-stage stochastic programs with access to a finite number of samples of random variables and concurrently observed covariates. Recently, Lin et al. (2022) has applied a conditional VaR constrained CSO formulation to the newsvendor problem. While most of the related work focuses on an “estimate-then-optimize” approach (see also Srivastava et al. (2021) and Hu, Kallus, and Mao (2022)), there have also been recent efforts in designing CSO models using an end-to-end paradigm (see Elmachtoub and Grigas (2022) and Donti, Amos, and J. Kolter (2017)).

Distributionally robust CSO One common challenge with the applications of CSO is due to the fact that often there are only a few samples (if any at all) drawn from the conditional distribution of ξ given ψ for each realization of ψ (Hu et al. 2020). This in turn causes a poor approximation of the true conditional distribution resulting in poor out-of-sample performance. Most proposed solutions to this issue have relied on distributionally robust optimization (DRO). For example, Bertsimas and Van Parys (2021), Bertsimas, McCord, and Sturt (2022) and Nguyen et al. (2021), and Srivastava et al. (2021) all propose DRO approaches that employ distribution sets that are centered at either the estimated conditional distribution or joint empirical distribution of (ψ, ξ). Kannan, Bayraksan, and J. Luedtke (2021) applies distributionally robust optimization to the residual-based CSO model proposed in Kannan, Bayraksan, and J. R. Luedtke (2020). We finally note that none of these works have considered the problem of conditional DRO where the distributional ambiguity set itself, namely its support or size, depending on contextual information.

Data-driven Robust Optimization and One-class Classification There has been a growing set of papers (see Ohmori (2021), C. G. McCord (2019), and Wang and Jacquillat (2020)) proposing various frameworks that use both supervised and unsupervised one-class classification techniques in designing the uncertainty sets which are further integrated into the RO problems. Some approaches make use of variance and covariance of historical data (Natarajan, Pachamanova, and Sim 2008) while others (Goerigk and Kurtz 2020; C. Wang et al. 2021) have exploited the representative power of deep neural networks to construct compact uncertainty sets. Up to this day, none of the data-driven robust optimization approaches have considered accounting for contextual information.

Deep Clustering Methods Traditional clustering methods like Gaussian Mixture Models (GMM) and *k*-means clustering rely on the original data representations and suffer from the curse of dimensionality. Recent developments in DNNs led to the learning of high quality representations, especially auto-encoder (AE) and decoder systems are particularly appealing as they are able to learn the representations in a fully unsupervised fashion. Several works like Chang et al. (2017), X. Guo et al. (2017), and Ji et al. (2017) combine variational AEs and GMMs to perform clustering and non-linearly map the input data into a latent space. Few works like Fard, Thonet, and Gaussier (2020) try to jointly learn the representations and jointly cluster with *k*-means and learning representations. We modify these algorithms to introduce a probability simplex that interacts with the centroids and also the center of the uncertainty sets.

1.3 The Deep Data-Driven Robust Optimization (DDDRO) Approach

Focusing on a classical robust optimization model, i.e. $\min_{x\in\mathscr{X}}\max_{\xi\in\mathscr{U}} c(x,\xi)$, the authors of Goerigk and Kurtz (2020) propose to employ deep learning to characterize the uncertainty set $\mathscr{U}$ in a data-driven environment. In particular, they consider describing the uncertainty set $\mathscr{U}$ in the form:

$$\mathscr{U}(W,R) := \{\ \xi \in \mathbb{R}^m : \|f_W(\xi) - \bar{f}_0\| \leq R\}, \tag{1.4}$$

where $f_W : \mathbb{R}^m \to \mathbb{R}^d$ is a deep neural network, parametrized using W, that projects the perturbation vector ξ to a new vector space where the uncertainty set can be more simply defined as a sphere of radius R centered at some $\bar{f}_0$.

Given a dataset $\mathscr{D}_\xi = \{\xi_1, \xi_2 \dots \xi_N\}$, they propose discovering the underlying structure of $\mathscr{U}$ by training the NN using a method found in the one-class classification literature, namely minimizing the empirical centered total variation of the projected data points:

$$\min_W \frac{1}{N}\sum_{i=1}^{N} \|f_W(\xi_i) - \bar{f}_0\|^2, \tag{1.5}$$

where $\bar{f}_0 := (1/N)\sum_{i\in[N]} f_{W_0}(\xi_i)$ is the center of the projected points under some initial random choice of f_{W_0}. Once the network is trained, they calibrate the radius R of $\mathscr{U}$ in order to reach a targeted coverage $1-\varepsilon$ of the data set.

In terms of NN architecture, they favor a special class of fully connected neural networks of depth L:

$$f_W(c) = \sigma^L(W^L\sigma^{L-1}(W^{L-1}\dots\sigma^1(W^1(c))\dots)) \tag{1.6}$$

where each W^ℓ captures a linear projection while each σ^ℓ captures a term-wise piecewise linear activation function (e.g. ReLU, Hardtanh, or hard sigmoid):

$$\sigma_j^\ell(w_j) = a_k^\ell w_j + b_k^\ell \text{ if } \underline{\alpha}_k^\ell \le w_j \le \overline{\alpha}_k^\ell, \quad k = 1, \dots, K$$

with $\{a_k^\ell, b_k^\ell, \underline{\alpha}_k^\ell, \overline{\alpha}_k^\ell\}_{k=1}^K$ as the parameters that identifies each of the K affine pieces.

The motivation for such an architecture comes from the proposed solution scheme for the RO problem, which relies on a constraint generation approach (see Algorithms 3 and 4 in Appendix). This scheme relies on progressively adding scenarios to a reduced set $\mathscr{U}' \subseteq \mathscr{U}$ until the worst-case cost of the solution under $\mathscr{U}'$ is the same as under $\mathscr{U}$. Numerically, a critical step consists in identifying the worst-case realization in $\mathscr{U}$, which is shown to reduce to a mixed-integer linear program when $c(x,\xi)$ is linear in ξ under the selected NN architecture due to the following representation of $\mathscr{U}(W,R)$:

$$\mathscr{U}(W,R) = \left\{ \xi \;\middle|\; \begin{array}{c} \exists u \in \{0,\,1\}^{d\times K\times L},\ \zeta \in \mathbb{R}^{d\times L},\ \phi \in \mathbb{R}^{d\times L} \\ \sum_{k=1}^K u_j^{k,\ell} = 1,\ \forall j,\ell \\ \phi^1 = W^1\xi \\ \zeta_j^\ell = \sum_{k=1}^K u_j^{k,\ell} a_k^\ell \phi_j^\ell + \sum_{k=1}^K u_j^{k,\ell} b_k^\ell,\ \forall j,\ell \\ \phi^\ell = W^\ell \zeta^{\ell-1},\ \forall \ell \ge 2 \\ \sum_{k=1}^K u_j^{k,\ell} \underline{\alpha}_k^\ell \le \phi_j^\ell \le \sum_{k=1}^K u_j^{k,\ell} \overline{\alpha}_k^\ell,\ \forall j,\ell \\ \|\zeta^L - \bar{f}_0\| \le R \end{array} \right\}, \tag{1.7}$$

where we assume for simplicity that each layer of the deep neural network has d neurons and ϕ^ℓ is the output at l-th layer of the neural network. We refer interested readers to Goerigk and Kurtz (2020) for more details.

1.4 Deep Data-driven Conditional Robust Optimization

Let (ψ, ξ) be a pair of random vectors defining respectively the side-information and random perturbation vectors of a contextual optimization problem. We can call our dataset $\mathscr{D}_{\psi\xi} := \{(\psi_1, \xi_1), \ldots, (\psi_N, \xi_N)\}$. Our objective is to train a data-driven conditional uncertainty set $\mathscr{U}(\psi)$ that will lead to robust solutions that are adapted to the type of perturbance that is experienced when ψ is observed. In this section, we propose two algorithms, namely the Deep cluster then classify (DCC) and the Integrated Deep cluster then classify (IDCC), to do so, and propose a calibration procedure that offers some guarantees with respect to a contextual value-at-risk problem.

1.4.1 The Deep "Cluster then Classify" (DCC) Approach

A direct extension of G&K's DDDRO approach in Section 1.3 consists in reducing the side-information ψ to a set of K different clusters, which provides states of the environment in which one wishes to design customized data-driven uncertainty sets. Mathematically, $\mathscr{U}(\psi) := \mathscr{U}_{a(\psi)}$, where $a : \mathbb{R}^m \to [K]$, is a trained K-class cluster assignment function for ψ, and each $\mathscr{U}_k$, for $k = 1, \ldots, K$, is an uncertainty sets for ξ that is trained and sized using the procedure described in Section 1.3 with the dataset $\mathscr{D}_{\xi}^{k} := \cup_{(\psi,\xi)\in\mathscr{D}_{\psi\xi}:a(\psi)=k}\{\xi\}$. This process implicitly involves multiple sequential steps of training deep neural networks. Following Moradi Fard, Thonet, and Gaussier (2020), when performing deep K-mean clustering to obtain $a(\psi)$, training can take the form of Algorithm 5, where the deep K-means algorithm trains simultaneously a representation $g_{V_E} : \mathbb{R}^m \to \mathbb{R}^d$, using an encoder and $g_{V_D} : \mathbb{R}^d \to \mathbb{R}^m$, using a decoder network, and a K-mean classifier $\bar{a}^{\theta}(\phi) := \operatorname{argmin}_{k\in[K]} \|\phi - \theta^k\|_2$ by minimizing, using stochastic gradient descent in a coordinate descent scheme, a trade-off (using α_K) between reconstruction error and the within cluster centered total variation in the encoded space:

$$\mathscr{L}^1(V, \theta) := (1 - \alpha_K)\frac{1}{N}\sum_{i=1}^{N}\|g_{V_D}(g_{V_E}(\psi_i)) - \psi_i\|^2 + \alpha_K\frac{1}{N}\sum_{i=1}^{N}\|g_{V_E}(\psi_i) - \theta^{a(\psi_i)}\|^2, \quad (1.8)$$

where $a(\psi) := \bar{a}^\theta(g_{V_E}(\psi))$. To solve this problem, we iterate between improving $V := (V_E, V_D)$ while keeping θ fixed, and improving θ while preserving V fixed.

Once the K-mean and one-class classifiers are trained, we correct for a deficiency of DDDRO approach, which assumes wrongfully that the projected $f_{W^k}(\xi)$ are normalized for each $\mathscr{D}_\xi^k$. Namely, we replace $\mathscr{U}(W,R)$ with a set that employs an ellipsoid in the projected space according to the statistics of $\mathscr{D}_\xi^k$:

$$\mathscr{U}(W^k, R^k, \mathscr{S}^k) := \{\xi \in \mathbb{R}^m : \|{\Sigma_f^k}^{-1/2}(f_{W^k}(\xi) - \mu_f^k)\| \leq R^k\}, \tag{1.9}$$

where $\mathscr{S}^k$ is short for (μ_f^k, Σ_f^k) with

$$\mu_f^k := |\mathscr{D}_\xi^k|^{-1} \sum_{\xi \in \mathscr{D}_\xi^k} f_{W_0^k}(\xi) \text{ and } \Sigma_f^k := |\mathscr{D}_\xi^k|^{-1} \sum_{\xi \in \mathscr{D}_\xi^k} (f_{W^k}(\xi) - \mu^k)(f_{W^k}(\xi) - \mu^k)^T .$$

The calibration of each R^k can finally be done using the same procedure as in Goerigk and Kurtz (2020) but using the reduced dataset $\mathscr{D}_\xi^k$.

1.4.2 The Integrated Deep Cluster-Classify (IDCC) Approach

While the simplicity of the approach presented in Section 1.4.1 makes it appealing, we identify two important weaknesses. First, by separating the training into multiple steps, it omits tackling the conditional uncertainty set learning problem as a whole. Namely, that low total variation in the ψ space (or a projection of it) does not necessarily imply that low total variation can easily be achieved in a projection of the ξ space. Second, it is unclear how to adapt the approach to a context where a clear separation of the clusters is impossible and where the notion of partial membership to a cluster is more appropriate.

To address the first problem, we propose an integrated framework for performing deep clustering and deep uncertainty set design jointly. Namely, we propose to optimize all of V, θ, and $\{W^k\}_{k=1}^K$ jointly using a loss function that trades-off between the objectives used for clustering and each of the K versions of one-class classifiers. We also tackle the issue of hard assignments by training a parameterized random assignment policy $\pi : \mathbb{R}^m \to \Delta_K$, where Δ_K is the probability simplex in $\mathbb{R}^K$, and θ the parameters that define the policy space. In the context of employing a soft version of deep K-means (Fard, Thonet, and Gaussier 2020), this random assignment policy takes the form of $\pi(\psi) := \bar{\pi}^{\theta}(g_V(\psi))$, where

$$\bar{\pi}_k^{\theta}(\psi) := \frac{\exp\{-\beta\|g_V(\psi) - \theta^k\|^2\}}{\sum_{k'=1}^{K} \exp\{-\beta\|g_V(\psi) - \theta^{k'}\|^2\}} \tag{1.10}$$

With these adjustments, our proposed loss function takes the form of:

$$\begin{aligned}\mathscr{L}_{\alpha}^{3}(V, \theta, \{W^k\}_{k=1}^{K}) :=& \alpha_S\Big((1-\alpha_K)\mathbb{E}_{\mathscr{D}}^{\pi}[\|g_{V_D}(g_{V_E}(\psi_i)) - \psi_i\|^2] \\ &\qquad + \alpha_K \mathbb{E}_{\mathscr{D}}^{\pi}[\mathrm{TotalVar}_{\mathscr{D}}^{\pi}(g_{V_E}(\psi), \theta^{\tilde{a}(\psi)}\,|\tilde{a}(\psi))]\Big) \\ &\qquad + (1-\alpha_S)\frac{1}{K}\sum_{k=1}^{K} \min_{\vartheta^k} \mathrm{TotalVar}_{\mathscr{D}}^{\pi}(f_{W^k}(\xi), \vartheta^k\,|\tilde{a}(\psi) = k), \end{aligned} \tag{1.11}$$

where $\tilde{a}(\psi) \sim \bar{\pi}^{\theta}(g_{V_E}(\psi))$ is the randomized assignment based on ψ, $\mathrm{TotalVar}_{\mathscr{D}}^{\pi}(\phi, \theta|\tilde{a}(\psi))$ $:= \sum_{j=1}^{d} \mathbb{E}_{\mathscr{D}}^{\pi}[(\phi_j - \theta_j)^2|\tilde{a}(\psi)]$ is the conditional centered total variation of given $\tilde{a}(\psi)$. In fact, all statistics are measured using the empirical distribution expressed in $\mathscr{D}_{\psi\xi}$ and the conditional distribution produced by the randomized assignment policy $\bar{\pi}^{\theta}(g_V(\psi))$, i.e. $\mathbb{P}_{\mathscr{D}}^{\pi}((\psi, \xi, \tilde{a}) \in \mathscr{E}) = (1/N)\sum_{i=1}^{N}\sum_{k=1}^{K} \mathbf{1}\{(\psi_i, \xi_i, k) \in \mathscr{E}\}\bar{\pi}_k^{\theta}(g_V(\psi_i))$. The explicit form of equation (1.11) can be found in Appendix 1.7.2.

Overall, $\mathscr{L}_{\alpha}^{3}$ trades off (using α_S) between the reconstruction error of the encoder-decoder networks on ξ, the expected recognizability of the K clusters, i.e. the fact that the observed features $g_{V_E}(\psi)$ form distinct clusters of points, and the average compactness of the produced conditional uncertainty sets. In particular, as $\alpha_S \to 1$, we can expect the minimizer of $\mathscr{L}_{\alpha}^{3}$ to converge to the minimizer of the cluster and classify approach. At the other end of the spectrum, when $\alpha_S \to 0$, the model will produce more self contained conditional uncertainty sets but at the price of less distinguishable clusters (in terms of ψ) that might poorly exploit the side-information. Algorithm 1 presents our proposed training scheme for the IDCC approach.

Given that we employ a random assignment policy, we propose replacing the deterministic CRO problem with its randomized version:

$$\tilde{\boldsymbol{x}}^*(\psi) \in \underset{x \in \mathscr{X}}{\operatorname{argmin}} \max_{\xi \in \tilde{\mathscr{U}}(\psi)} c(x, \xi),$$

where $\tilde{\mathscr{U}}(\psi) := \mathscr{U}(W^{\tilde{a}(\psi)}, R^{\tilde{a}(\psi)}, \mathscr{S}^{\tilde{a}(\psi)})$[1] is a random uncertainty set, and where we express the fact that conditionally on ψ, $\tilde{\boldsymbol{x}}(\psi)$ is a random policy that depends on the realization of $\tilde{a}$. Given the randomness of $\tilde{\mathscr{U}}(\psi)$, one needs to be more careful in defining a calibration scheme for each R^k. Our proposed scheme is motivated by the following lemma, which proof can be found in Appendix 1.7.1.

Lemma 1.4.1. *Let the random uncertainty set $\tilde{\mathscr{U}}(\psi)$ satisfy:*

$$\mathbb{P}^{\pi}_{\mathscr{D}}(\xi \in \tilde{\mathscr{U}}(\psi) | \tilde{a}(\psi) = k) \geq 1 - \varepsilon, \forall k, \tag{1.12}$$

then it satisfies:

$$\mathbb{P}^{\pi}_{\mathscr{D}}(\xi \in \tilde{\mathscr{U}}(\psi)) \geq 1 - \varepsilon. \tag{1.13}$$

In particular, this lemma suggests calibrating each R^k using the bisection to solve:

$$\inf \left\{ R \middle| \frac{\sum_{i=1}^N \mathbf{1}\{\xi_i \in \mathscr{U}(W^k, R, \mathscr{S}^k)\} \bar{\pi}_k^{\theta}(g_{V_E}(\psi_i))}{\sum_{i=1}^N \bar{\pi}_k^{\theta}(g_{V_E}(\psi_i))} \geq 1 - \varepsilon \right\}, \tag{1.14}$$

given that the resulting R^k are the smallest that satisfy (1.12).

[1] Here, $\mathscr{S}^k$ refers to $(\bar{f}^{\theta,V}_{W^k|\tilde{a}(\psi_i)=k}, \bar{\Sigma}^{\theta,V}_{W_k|\tilde{a}(\psi)=k})$ with

$$\bar{\Sigma}^{\theta,V}_{W_k|\tilde{a}(\psi)=k} := \sum_{i=1}^{N} \frac{\bar{\pi}_k^{\theta}(g_{V_E}(\psi_i))}{\sum_{i=1}^N \bar{\pi}_k^{\theta}(g_{V_E}(\psi_i))} \cdot (f_{W^k}(\xi_i) - \bar{f}^{\theta,V}_{W^k|\tilde{a}(\psi_i)=k})(f_{W^k}(\xi_i) - \bar{f}^{\theta,V}_{W^k|\tilde{a}(\psi_i)=k})^T$$

.

Algorithm 1 : Integrated deep cluster-classify with deep K-means

```
Input : Dataset $\mathscr{D}_{\xi,\psi}$; number of clusters $K$; hyperparameters $\alpha_{\mathrm{K}}, \alpha_{\mathrm{S}}, \beta$
Randomly initialize $\theta_0$, $V_0$, and $W_0$
Let $\pi_0 := \bar{\pi}^{\theta_0}(g_{V_{E0}}(\psi))$ and $W_0^k := W_0$ for all $k$'s
Set $t := 0$
repeat
    Set $t := t+1$
    Update $\theta_t^k := \mathbb{E}_{\mathscr{D}}^{\pi}[g_{V_{Et-1}}(\psi) \mid \tilde{a}(\psi) = k]$ using $\pi_{t-1}$
    Update $(V_t, \{W_t^k\}_{k=1}^K)$ using gradient descent on Eq. (1.11) with $\theta_t$
    Get $\pi_t := \bar{\pi}^{\theta_t}(g_{V_{Et}}(\psi))$
until $t \geq T$ or convergence
Let $\pi(\cdot) := \pi_t(\cdot)$ and $W^k := W_t^k$ for all $k$
for $k = 1$ to $K$ do
    Calibrate $R^k$ using Eq. (1.14)
    Let $\mathscr{U}^k := \mathscr{U}(W^k, R^k, \mathscr{S}^k)$
return $\pi(\cdot)$ and $\{\mathscr{U}^k\}_{k=1}^K$
```

1.4.3 Connections to Contextual Value-at-Risk Optimization

In the previous subsections, we proposed two different schemes to produce a possibly randomized uncertainty set $\tilde{\mathscr{U}}(\psi)$ that can be employed in a randomized CRO problem[2]. We also proposed a scheme for radii calibration so that they would satisfy the coverage property in equation (1.13). Hence, one can derive the following connection between conditional robust optimization and the CVO Problem (1.1). The proof is pushed to Appendix 1.7.1.

Lemma 1.4.2. *When $\tilde{\mathscr{U}}$ satisfies (1.13), the random policy $\tilde{\mathbf{x}}(\cdot)$ to the randomized CRO*

[2] Note that in the case of Section 1.4.1, the conditional uncertainty set is deterministic thus reducing the randomized version of CRO to a pure CRO problem

problem together with

$$v^* := esssup^{\pi}_{\mathscr{D}} \min_{x\in\mathscr{X}} \max_{\xi\in\tilde{\mathscr{U}}(\psi)} c(x,\xi)$$

provide a conservative approximate solution to the CVO problem under the empirical measure $\mathbb{P}^{\pi}_{\mathscr{D}}$. Namely,

$$VaR^{\mathscr{D},\pi}_{1-\varepsilon}(c(\tilde{\boldsymbol{x}}(\psi),\xi)) \leq v^*.$$

In particular, in the case of the proposed DCC and IDCC approaches we have that

$$v^* = \max_{k\in[K]} \min_{x\in\mathscr{X}} \max_{\xi\in\mathscr{U}(W^k,R^k.\mathscr{S}^k)} c(x,\xi)\,.$$

As the robust optimization paradigm traditionally aims at offering statistical guarantees on the out-of-sample performance of the prescribed solutions, we describe below how a bootstrap method can be used to estimate the radii R^k's.

Remark 1.4.1. *Using bootstrapping methods, we can get a conservative approximation of each R_k as:*

$$\tilde{R}_k := \inf\left\{ R \,\middle|\, \mathbb{P}_{\tilde{\mathscr{D}}}\left(\sum_{i=1}^{N} \frac{\bar{\pi}^{\theta}_{k}(g_{V_E}(\psi_i)}{\sum_{i=1}^{N} \bar{\pi}^{\theta}_{k}(g_{V_E}(\psi_i)} \mathbf{1}\{\xi_i \in \mathscr{U}(W^k,R,\mathscr{S}^k)\} \geq 1-\varepsilon\right) \;\geq\; 1-\delta\right\}$$

where $\mathbb{P}_{\tilde{\mathscr{D}}}$ measures the probabibility when resampling a new dataset of size N with replacement from $\mathscr{D}_{\psi\xi}$. When N is large enough and assuming that each data point is drawn i.i.d. according to some unknown probability measure $\mathbb{P}$, we asymptotically get the guarantee that $\mathbb{P}(\xi\in\tilde{\mathscr{U}}(\psi)) \geq 1-\varepsilon$ with probability higher than approximately $1-K\delta$.

1.5 Experiments

In this section, we illustrate the coverage aspect of the IDCC approach using simulated data. We will further demonstrate the advantage of the CRO problem using a standard risk minimizing portfolio optimization problem. We compare the performance of IDCC with that of DCC, DDDRO (with ellipsoidal correction in (1.9)), and the classical ellipsoidal uncertainty approach (i.e. DCC with $K = 1$ and $f_{W^1}(\xi) := \xi$). The IDCC and DCC methods incorporate the covariate information whereas DDDRO and ellipsoid approaches ignore this information. The neural network architecture and other modeling information are available in Appendix 1.7.2. The code can be found on github[3]. Our code uses the Pytorch implementation from Goerigk and Kurtz (2020), which is available online[4].

1.5.1 Conditional Uncertainty Set Illustration Using Simulated Data

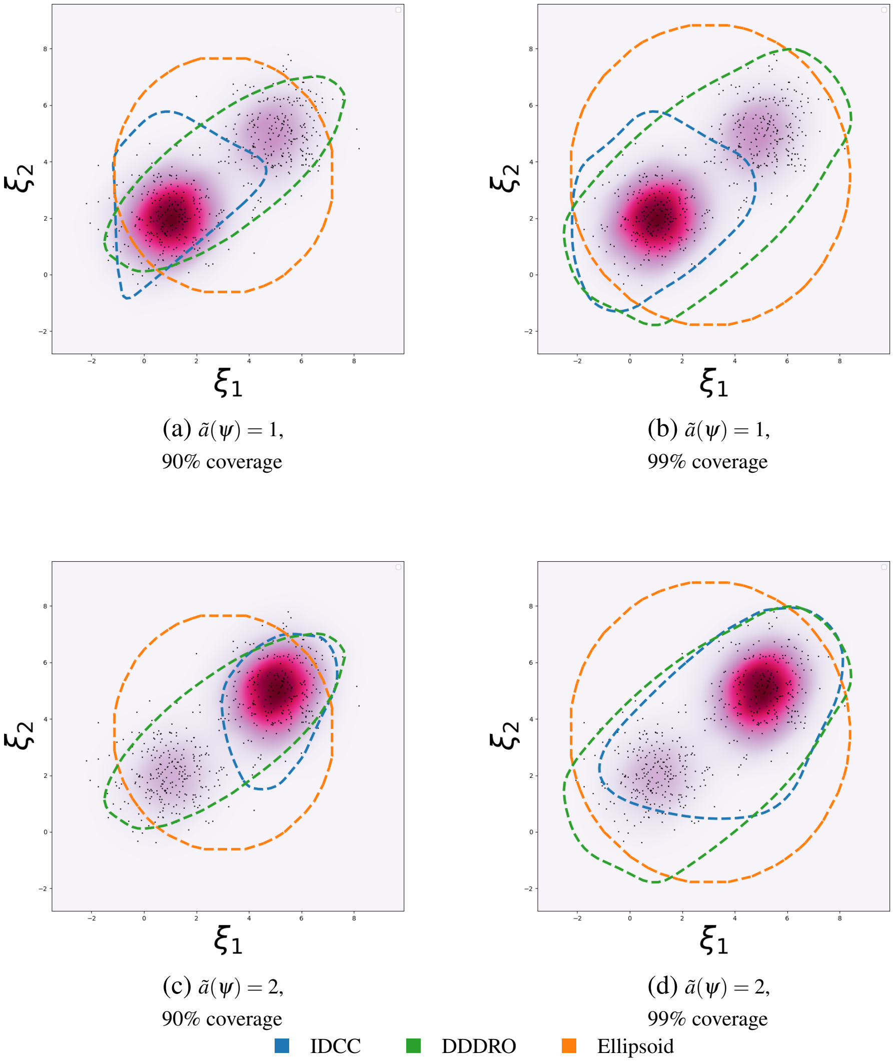
For ease of illustration, we consider a simulation environment where $[\psi^T \;\; \xi^T] \in \mathbb{R}^4$ is a random vector whose distribution is an equal-weighted mixture of two 4-d multivariate normal distributions. We consider $N = 500$ and train IDCC (with $K = 2$), DDDRO, and the ellipsoid and calibrate the uncertainty sets for a probability coverage of 90%, 99% (i.e. $\varepsilon \in \{1\%,\ 10\%\}$). As a result, DDDRO and IDCC, which use deep neural networks, identify non-convex uncertainty sets, whose convex hulls are presented in Figure 1.1 together with the calibrated ellipsoid.The figure also presents the conditional distribution of ξ according to $\mathbb{P}^{\pi}_{\mathscr{D}}(\cdot|\tilde{a}(\psi) = k)$, using IDCC's randomized assignment, and the training dataset. One can remark that the conditional sets produced by IDCC exploit the side information by concentrating the uncertainty set on the region that has the most mass according to $\mathbb{P}^{\pi}_{\mathscr{D}}(\cdot|\tilde{a}(\psi) = k)$ thus leading to a less conservative RO problem then DDDRO and the ellipsoid, which are oblivious to ψ. In fact, it appears to have successfully learned to at least partially recognize the mixture membership using ψ and exploit this information to adapt the uncertainty set.

1.5.2 Robust Portfolio Optimization

We further investigate the empirical out-of-sample performance of the proposed uncertainty sets on a classical robust portfolio optimization problem. Namely, we consider a situation where an investor is trying to minimize the worst-case return based on an uncertainty set that provides $1 - \varepsilon$ probabilistic coverage of the uncertain future return vector. In particular, given that x captures a vector of investment in $n = m$ different assets whose

[3] https://anonymous.4open.science/r/Data-Driven-Conditional-Robust-Optimization-E160/

[4] https://github.com/goerigk/RO-DNN



(a) $\tilde{a}(\psi) = 1$,
90% coverage

(b) $\tilde{a}(\psi) = 1$,
99% coverage

(c) $\tilde{a}(\psi) = 2$,
90% coverage

(d) $\tilde{a}(\psi) = 2$,
99% coverage

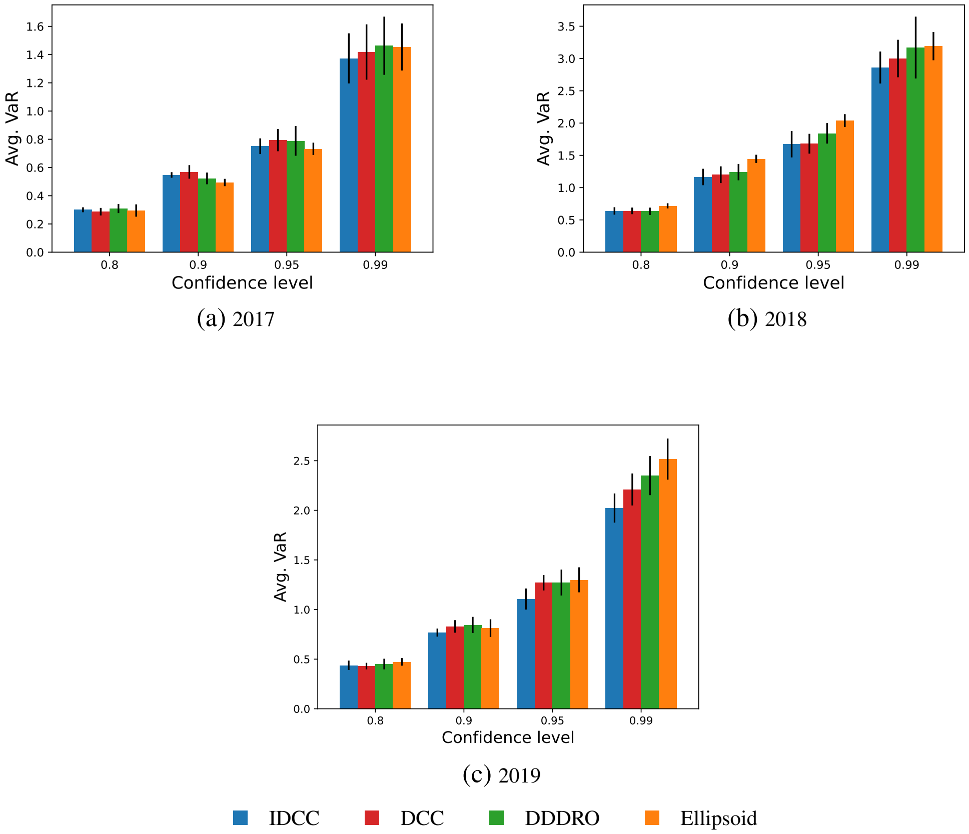
Figure 1.1: Convex hull of trained uncertainty sets for two levels of coverage and with a conditional uncertainty set for IDCC that exploits two clusters. The heatmap represents the conditional distribution of ξ according to $\mathbb{P}^{\pi}_{\mathscr{D}}(\cdot|\tilde{a}(\psi) = k)$. The cloud of points represents the training dataset.

return are captured using ξ, we let $c(x, \xi) := -\xi^\intercal x$ to capture the return on investment, and let $\mathscr{X} := \{x \in \mathbb{R}^n | \sum_{i=1}^n x_i = 1,\ x \geq 0\}$ to capture the need to invest one unit of wealth among the available assets. Following Lemma 1.4.2, this model can in turn be interpreted as conservatively approximating a $\min_{x \in \mathscr{X}} \mathrm{VaR}_{1-\varepsilon}(\xi^\intercal x)$, where the objective is a risk averse value-at-risk metric.

Dataset Our experiments make use of historical data from the U.S. stock market. We collect the adjusted daily closing prices for 70 stocks (as used in Xu and Cohen (2018)) coming from 8 different sectors from January 1, 2012, to December 31, 2019, using the Yahoo! Finance's API. Each year has 252 data points and we compute the percentage gain/loss w.r.t the previous day to create our dataset for ξ. As for side information, we use the trading volume of individual stocks and other market indices[5] over the same period as covariates. Our algorithm gives the flexibility to use any number of such metrics as contextual information. Given the time series nature of the data, at a given instance, we use 3 years of data to train and the following year as validation to pick the hyperparameters of our model such as learning rate, weight decay, and the optimal number of clusters. We then retrain the model using the 4 years of data to build the final model. Upon calibrating the uncertainty set, we use it to solve the robust portfolio optimization problem. We then apply this policy to the next 1 year's of data and compute the performance metric, namely Value at risk (VaR) for different confidence levels to compare the performances. VaR quantifies the level of risk of a portfolio over a specified time frame. Here, it gives an estimate of the maximum % loss the decision maker can incur over a period of 1 year when he uses the policy from the RO model. Intuitively, lower the VaR, less riskier is the generated policy. Many financial institutions use VaR to determine the amount of collateral needed when trading financial products so lowering VaR for high confidence levels is crucial.

Experiment Design To test for the robustness of the IDCC algorithm, we experiment on various randomly sampled stock combinations across different time periods. We randomly sampled a subset of 15 stocks in a time window and repeated the experiment

[5] Volatility Index (VIX), 10-year Treasury Yield Index (TNX), Oil Index (CL=F), S&P 500 (GSPC), Global Income & Currency Fund (XGCFX), Dow Jones Index (DJI)



(a) 2017

(b) 2018

(c) 2019

Figure 1.2: Avg. VaR across portfolio simulations. Error bars report 95% CI.

for 10 runs on 3 moving time frames. We used learning rate = 0.01, $\alpha_K = 0.5, \alpha_S = 0.5$, $\beta = 0.1$ for all the experiments. We use a cold start K-means approach to determine K for each run. We do this across all these experiments as it will be computationally expensive to tune the parameters through grid search for each run and also our intention is to show the learning capability of our algorithm even with minimal tuning. The parameter tuning and implementation details can be found in Appendix 1.7.2.

Results Figure 1.2 shows the avg. VaR across the runs at different confidence levels. It is evident that IDCC generally performs better than the baseline models. This difference is especially noticeable at a higher confidence level and vanishes as we move to lower confidence levels. Table 1.1 provides more details by comparing the overall and conditional cluster level VaR with the baseline models. Specifically, in each run, we identify each cluster as either the “majority” or “minority” cluster depending on its frequency and report averages of VaR (among the 10 runs) for each of these labels. The average frequencies for each label are also reported in the table. In particular, one can observe that the improvement on average overall VaR can reach up to $\sim$15% (see in 2019 at a 0.99 confidence level). This advantage is even more clearly visible when we look at the individual cluster-level conditional VaR. For instance, in the year 2018 for the 0.99 confidence level, the majority cluster ($\sim$68% data) provides an improvement of 19% and an overall improvement of 9% compared to the second best baseline model. A similar pattern is observed for the year 2019 as well. In the year 2017, the overall performance of IDCC is close and for some confidence levels slightly above the baseline models. However, we see that the majority cluster ($\sim$80% data) is performing better than the baseline models while the minority cluster has a slightly higher risk. We attribute this loss in performance to the fact that the minority clusters are much less frequent ($\sim$20% data) and therefore have fewer data available to properly learn its conditional uncertainty set. This large difference in frequencies might also indicate that the side information does not have a strong signal for the behavior of the returns during this period of time.

		2017				2018				2019			
	Conf. $1-\varepsilon$	0.8	0.9	0.95	0.99	0.8	0.9	0.95	0.99	0.8	0.9	0.95	0.99
Overall	IDCC	0.30	0.55	0.75	**1.37**	0.64	**1.16**	**1.67**	**2.86**	**0.44**	**0.77**	**1.11**	**2.02**
	DDDRO	0.31	0.52	0.79	1.46	**0.63**	1.24	1.84	3.17	0.45	0.84	1.27	2.35
	Ellipsoid	0.30	**0.49**	0.75	1.45	0.72	1.45	2.04	3.19	0.47	0.81	1.30	2.52
Cond. on Majority Cluster	Cluster Freq.	80%				68%				59%			
	IDCC	0.31	0.52	**0.71**	**1.30**	**0.57**	**1.08**	**1.50**	**2.62**	**0.44**	**0.75**	**1.17**	**1.88**
	DDDRO	0.31	0.52	0.74	1.35	0.59	1.15	1.63	3.23	0.45	0.85	1.31	2.06
	Ellipsoid	0.32	0.52	0.74	1.41	0.69	1.29	1.92	3.08	0.47	0.85	1.25	2.31
Cond. on Minority Cluster	Cluster Freq.	20%				32%				41%			
	IDCC	0.30	0.61	0.77	1.43	**0.96**	**1.57**	2.05	**3.13**	**0.48**	0.82	**1.15**	**2.22**
	DDDRO	0.30	0.56	0.84	1.39	1.00	1.66	**2.04**	3.30	0.49	0.84	1.40	2.39
	Ellipsoid	**0.28**	**0.47**	**0.69**	**1.13**	1.17	1.80	2.43	3.43	0.49	0.82	1.38	2.57

Table 1.1: Comparison of average value-at-risk (over 10 runs) for different levels of probability coverage. Both the overall VaR and conditional VaR given the membership to the majority/minority clusters are presented.

1.6 Conclusion and Future Work

In this work, we introduced a new approach, Conditional Robust Optimization, for solving contextual optimization problems in a risk averse setting. We proposed a novel integrated approach to design uncertainty sets that adapt to revealed covariate information. We identified connections to contextual value-at-risk optimization and showed empirically that our method reduces the out-of-sample VaR considerably compared to non-contextual RO schemes when the level of protection needed is high. As future work, we find that it should be interesting to integrate data-driven conditional uncertainty sets in the context of multi-stage robust optimization models. Given that clustering techniques are often prone to capturing correlations that do not reflect true causal relations, a promising direction for future work is to integrate causal inference methods into our approach. One might also be concerned regarding fairness considerations in contexts where side information might allow to treat of a certain class of individuals differently from others. This last issue might be addressed by adding fairness consideration in our integrated loss function.

Acknowledgements

The authors gratefully acknowledge support from the Institut de Valorisation des Données (IVADO) and from the Canadian Natural Sciences and Engineering Research Council [Grant RGPIN-2016-05208 and 492997-2016].

1.7 Appendix

1.7.1 Proofs

As mentioned in Sections 3 and 4, our dataset $\mathscr{D}$ contains the random perturbation vectors ξ and side information ψ. $\tilde{\mathscr{U}}(\psi)$ represents the conditional uncertainty set that satisfies the following properties.

Lemma 1.4.1. *Let the random uncertainty set $\tilde{\mathscr{U}}(\psi)$ satisfy:*

$$\mathbb{P}^{\pi}_{\mathscr{D}}(\xi \in \tilde{\mathscr{U}}(\psi) | \tilde{a}(\psi) = k) \geq 1 - \varepsilon, \forall k \tag{1.15}$$

then it satisfies:

$$\mathbb{P}^{\pi}_{\mathscr{D}}(\xi \in \tilde{\mathscr{U}}(\psi)) \geq 1 - \varepsilon. \tag{1.16}$$

Proof. The claim follows from:

$$\begin{aligned} \mathbb{P}^{\pi}_{\mathscr{D}}(\xi \in \tilde{\mathscr{U}}(\psi)) &= \sum_{k=1}^{K} \mathbb{P}^{\pi}_{\mathscr{D}}(\xi \in \tilde{\mathscr{U}}(\psi) | \tilde{a}(\psi) = k) \mathbb{P}^{\pi}_{\mathscr{D}}(\tilde{a}(\psi) = k) \\ &\geq \sum_{k} (1 - \varepsilon) \mathbb{P}^{\pi}_{\mathscr{D}}(\tilde{a}(\psi) = k) = 1 - \varepsilon\,. \end{aligned}$$

□

Lemma 1.4.2. *When $\tilde{\mathscr{U}}$ satisfies (1.13), the random policy $\tilde{\mathbf{x}}(\cdot)$ to the randomized CRO problem together with*

$$v^* := \mathit{esssup}^{\pi}_{\mathscr{D}} \min_{x \in \mathscr{X}} \max_{\xi \in \tilde{\mathscr{U}}(\psi)} c(x, \xi)$$

provide a conservative approximate solution to the CVO problem under the empirical measure $\mathbb{P}^{\pi}_{\mathscr{D}}$. Namely,

$$\mathit{VaR}^{\mathscr{D},\pi}_{1-\varepsilon}(c(\tilde{\mathbf{x}}(\psi), \xi)) \leq v^*.$$

In particular, in the case of the DCC and IDCC approaches we have that

$$v^* = \max_{k \in [K]} \min_{x \in \mathscr{X}} \max_{\xi \in \mathscr{U}(W^k, R^k . \mathscr{S}^k)} c(x, \xi)\,.$$

Proof. First, by definition of $\tilde{\boldsymbol{x}}(\cdot)$ and v^*, we have that when $\xi \in \tilde{\mathscr{U}}(\psi)$:

$$c(\tilde{\boldsymbol{x}}(\psi),\xi) \le \max_{\xi\in\tilde{\mathscr{U}}(\psi)} c(\tilde{\boldsymbol{x}}(\psi),\xi) = \min_{x\in\mathscr{X}} \max_{\xi\in\tilde{\mathscr{U}}(\psi)} c(x,\xi) \le v^* .$$

Hence, we must have that:

$$\begin{aligned}\mathbb{P}^{\pi}_{\mathscr{D}}(c(\tilde{\boldsymbol{x}}(\psi),\xi) \le v^*) &\ge \mathbb{P}^{\pi}_{\mathscr{D}}(c(\tilde{\boldsymbol{x}}(\psi),\xi) \le v^* | \xi \in \tilde{\mathscr{U}}(\psi))\mathbb{P}^{\pi}_{\mathscr{D}}(\xi \in \tilde{\mathscr{U}}(\psi))\\ &\ge 1\cdot(1-\varepsilon)\ .\end{aligned}$$

We thus obtain our result based on the following argument:

$$\mathrm{VaR}^{\mathscr{D},\pi}_{1-\varepsilon}(c(\tilde{\boldsymbol{x}}(\psi),\xi)) \ := \inf\{t|\mathbb{P}^{\pi}_{\mathscr{D}}(c(\tilde{\boldsymbol{x}}(\psi),\xi) \le t) \ge 1-\varepsilon\} \ \le v^* .$$

In the case of the DCC and IDCC approaches we have that

$$v^* = \max_{k\in[K]} \min_{x\in\mathscr{X}} \max_{\xi\in\mathscr{U}(W^k,R^k)} c(x,\xi)\,,$$

since $\tilde{\mathscr{U}}(\psi)$ is supported on $\{\mathscr{U}(W^k,R^k,\mathscr{D}^{\pi}_{k})\}_{k=1}^{K}$. □

1.7.2 Deep Learning Implementation of IDCC Approach

IDCC Loss Function:

Mathematically, the conditional total variation loss function (1.11) can be explicitly written as:

$$\begin{aligned}&\mathscr{L}^3_{\alpha}(V,\theta,\{W^k\}_{k=1}^{K}) := (1-\alpha_S)\frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{N}\frac{\bar{\pi}_k^{\theta}(g_{V_E}(\psi_i))}{\sum_{i=1}^{N}\bar{\pi}_k^{\theta}(g_{V_E}(\psi_i))}\|f_{W^k}(\xi_i)-\bar{f}^{\theta,V}_{W^k|\tilde{a}(\psi_i)=k}\|^2\\ &+\alpha_S\Big((1-\alpha_K)\frac{1}{N}\sum_{i=1}^{N}\|g_{V_D}(g_{V_E}(\psi_i))-\psi_i\|^2+\alpha_K\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}\bar{\pi}_k^{\theta}(g_{V_E}(\psi_i))\|g_{V_E}(\psi_i)-\theta^k\|^2\Big)\end{aligned} \tag{1.17}$$

where

$$\bar{f}^{\theta,V}_{W^k|\tilde{a}(\psi_i)=k} := \sum_{i=1}^{N}\frac{\bar{\pi}_k^{\theta}(g_{V_E}(\psi_i))}{\sum_{i=1}^{N}\bar{\pi}_k^{\theta}(g_{V_E}(\psi_i))}f_{W^k}(\xi_i)\,.$$

IDCC Network Architecture:

The joint loss minimization task is performed using the following network architecture which has 2 parallel networks training simultaneously. The first network($g_V := (g_{V_E}, g_{V_D})$) takes the side information(ψ) as the input and generates a randomized assignment $\tilde{a}(\psi) \sim \bar{\pi}^\theta(g_{V_E}(\psi))$. The second network($\{f_{W^k}\}_{k=1}^K$) takes the random perturbation vector(ξ) and $\tilde{a}(\psi)$ as the input to generate $W^{\tilde{a}(\psi)}, \mathscr{S}^{\tilde{a}(\psi)}$[6] which are subsequently used to design the uncertainty set $\tilde{\mathscr{U}}(\psi) := \mathscr{U}(W^{\tilde{a}(\psi)}, R^{\tilde{a}(\psi)}, \mathscr{S}^{\tilde{a}(\psi)})$.

g_V is an auto-encoder (AE) network which generates the assignment vector $\tilde{a}(\psi)$. They are trained to learn lower dimension data representations at the bottleneck of the network. They have the capability to learn representations in a fully unsupervised way which makes them suitable for the task at hand. The encoder($g_{V_E}(.)$) consists of the input(dim=*m*), hidden and the output layers(dim=*d*). The decoder($g_{V_D}(.)$) uses this low dimension representation to reconstruct the original input data. The decoder is a mirrored version of the encoder. The input layer is fully connected to the output layers with an intermediate ReLU activation layer in both the encoder and the decoder. We initialize the network weights using kaiming normal initialization. The output from the encoder is passed through a softmax layer to generate a soft version of deep *K*-means Fard, Thonet, and Gaussier (2020) which gives the assignment simplex $\tilde{a}(\psi) \sim \bar{\pi}^\theta(g_{V_E}(\psi))$ where

$$\bar{\pi}_k^\theta(g_{V_E}(\psi)) := \frac{\exp\{-\beta \|g_{V_E}(\psi) - \theta^k\|^2\}}{\sum_{k'=1}^K \exp\{-\beta \|g_{V_E}(\psi) - \theta^{k'}\|^2\}} \tag{1.18}$$

The parallel network($\{f_{W^k}\}_{k=1}^K$) designs the *K* customized data-driven uncertainty sets using a slightly modified deep SVDD method from Goerigk and Kurtz (2020). The input to these networks is the perturbations ξ and the assignment policy($\bar{\pi}^\theta(g_{V_E}(\psi))$). Each f_{W^k} has an input layer(dim=15), hidden layer and an output layer(dim=5). All layers are fully connected with a ReLU activation function. All the networks are initialized with a uniform

[6] Here, $\mathscr{S}^k$ refers to $(\bar{f}^{\theta,V}_{W^k|\tilde{a}(\psi_i)=k}, \bar{\Sigma}^{\theta,V}_{W_k|\tilde{a}(\psi)=k})$ with

$$\bar{\Sigma}^{\theta,V}_{W_k|\tilde{a}(\psi)=k} := \sum_{i=1}^N \frac{\bar{\pi}_k^\theta(g_{V_E}(\psi_i))}{\sum_{i=1}^N \bar{\pi}_k^\theta(g_{V_E}(\psi_i))} \cdot (f_{W^k}(\xi_i) - \bar{f}^{\theta,V}_{W^k|\tilde{a}(\psi_i)=k})(f_{W^k}(\xi_i) - \bar{f}^{\theta,V}_{W^k|\tilde{a}(\psi_i)=k})^T,$$

.

distribution in $[0,1]$. Our approach constructs a weighted center, $\bar{f}^{\theta,V}_{W^k|\tilde{a}(\psi_i)=k}$ which uses $\bar{\pi}^{\theta}(g_{V_E}(\psi))$ to compute the loss in equation (1.17).

Suggested Extensive Parameter Tuning Procedure

In this section, we discuss the parameter tuning strategy that can be used to train the network proposed in Section 1.7.2 using the portfolio optimization example discussed in Section 1.5.2. Here, given the time series nature of the data, we follow the rolling window approach for network training. Our architecture uses a set of hyperparameters, $hp = (lr, \alpha_K, \alpha_S, \beta, K)$ where lr represents the learning rate, α_K regulates the trade-off between seeking good representations for ψ that are faithful to the original data and representations that are useful for clustering purposes. α_S plays a similar trade-off between the recognizability and compactness of uncertainty sets. Finally, β is a softmax temperature parameter and K represents the number of clusters. We split the data into training and validation periods and search for the optimal combination through the grid search method. For each combination, we train the network and generate the optimal policy using training data which is applied to the unseen validation data. The optimal combination is the one that gives the lowest $VaR_{1-\varepsilon}$ on the validation dataset as this is a worst case return minimization problem. This is shown in Algorithm 2. Once the hyperparameters are selected, we re-train the network using the complete data. It is important to note that the results reported in Section 1.5 did not use parameter tuning to reduce computations.

```
Algorithm 2 : Hyperparameter tuning
  Input : $hp = (lr, \alpha_{\mathrm{K}}, \alpha_{\mathrm{SV}}, \beta, K)$
  for $year = y$ to $y + M$ do
      Obtain $\{\mathscr{U}^k\}_{k=1}^{K}$ from Algorithm 1
      Get optimal portfolio using:
          $\min_{x \in \mathscr{X}} \mathrm{VaR}_{1-\varepsilon}(\xi^\intercal x)$        (see Section 1.5.2)
  Choose $hp$ which minimizes out-of-sample $\mathrm{VaR}_{1-\varepsilon}$ over $M$ periods
```

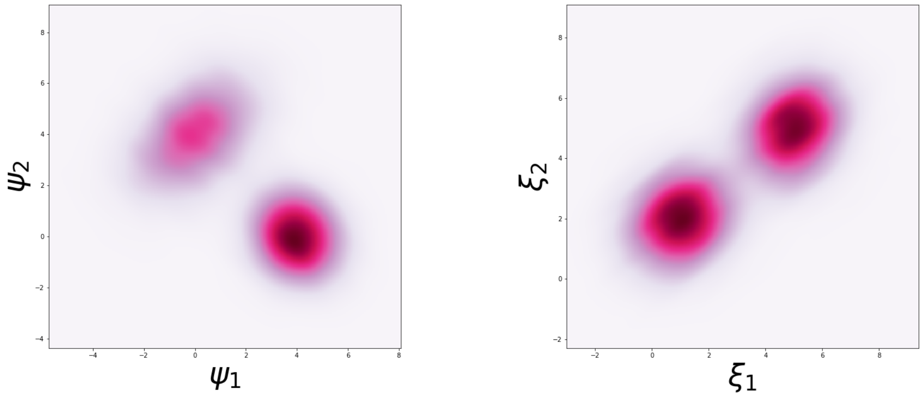
Simulated Data Generation Process

In this section, we discuss the data generation process for the simulated data used in Section 1.5.1. For easy visualization, we consider a simulation environment where $[\psi^T \ \ \xi^T]^T \in \mathbb{R}^4$ is a random vector whose distribution is an equal-weighted mixture of two 4-d multivariate normal distributions.[7] Namely, $[\psi^T \ \ \xi^T]^T \sim 0.5N(\mu_1, \Sigma_1) + 0.5N(\mu_2, \Sigma_2)$ where:

$$\mu_1 := \begin{bmatrix} 1 \\ 2 \\ 0 \\ 4 \end{bmatrix}, \qquad \Sigma_1 := \begin{bmatrix} 1.0 & 0.0 & 0.3 & -0.1 \\ 0.0 & 1.0 & 0.1 & -0.2 \\ 0.3 & 0.1 & 1.0 & 2.0 \\ -0.1 & -0.2 & 2.0 & 1.0 \end{bmatrix}$$

$$\mu_2 := \begin{bmatrix} 5 \\ 5 \\ 4 \\ 0 \end{bmatrix}, \qquad \Sigma_2 := \begin{bmatrix} 1.0 & 0.0 & 0.3 & -0.1 \\ 0.0 & 1.0 & 0.1 & -0.2 \\ 0.3 & 0.1 & 1.0 & 0.0 \\ -0.1 & -0.2 & 0.0 & 1.0 \end{bmatrix}.$$

The distribution marginalized over the random vectors $\psi \in \mathbb{R}^2$ and $\xi \in \mathbb{R}^2$ can respectively be visualized in Figure 1.3(a) and (b).



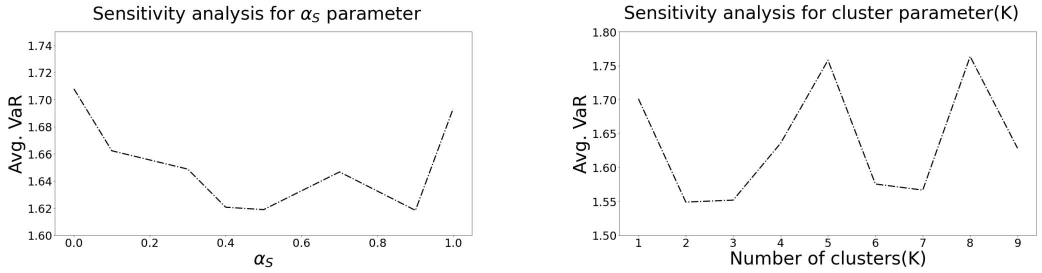
(a) (b)

Figure 1.3: Density plot of the marginalized distributions over ψ (in (a)) and ξ (in (b)) from a mixture of two Gaussian distributions on the joint space $[\psi^T \ \ \xi^T]^T$.

[7] The data is generated using Page Jr (1984).

Sensitivity Analysis for Parameters

Here, we show the sensitivity analysis for the parameters α_S and K. For each of these analyses, we keep all the other parameters constant and train the model by varying the considered parameters. For α_S, we consider the range of values between 0 and 1. For the sensitivity analysis of K, we considered 1 to 9 clusters. We conducted 10 such runs in the year 2019 and observe the average validation VaR. The results can be seen in the plots below. The analysis in Figure 1.4b shows that 2 clusters result in similar or improved performance compared to using more clusters. Regarding the influence of α_S on out-of-sample performance, we did not observe any insightful behavior. We believe this hyperparameter can play a role in problem settings where the convergence of TV losses in contextual and perturbed spaces is different and needs moderation. However, in this case, we don't notice any such issues and the choice of α_S as 0.5 seemed to work generally well across all experiments as seen in Figure 1.4a. The sensitivity analysis also highlights the same, which points to 0.5 as being a legitimate choice for α_S.



(a) (b)

Figure 1.4: Sensitivity analysis (using validation data) across portfolio simulations for the year 2019.

1.7.3 Algorithms

In this section, we provide the pseudo-code for the Iterative constraint generation and the Deep Cluster then Classify techniques from Section 1.4.1.

Iterative constraint generation

We present the iterative constraint generation algorithm for both the Robust Objective problem:

$$\min_{x \in \mathscr{X}} \max_{\xi \in \mathscr{U}} c(x, \xi),$$

and a Robust Constraint problem of the form:

$$\min_{x \in \mathscr{X} : c(x,\xi) \leq 0, \forall \xi \in \mathscr{U}} f(x).$$

We note that when $\mathscr{X}$ is convex and $c(x,\xi)$ is convex in x and linear in ξ, then $\arg\min_{x \in \mathscr{X}} \max_{\xi \in \mathscr{U}'} c(x,\xi)$ can be obtained using convex optimization algorithms, while $\xi^* \in \text{argmax}_{\xi \in \mathscr{U}} c(x^*, \xi)$ can be obtained using mixed-integer linear programming solvers such as MOSEK (see MOSEK ApS (2022)). In more general setting, one might need to employ more general non-linear programming software.

Algorithm 3 : Iterative constraint generation for robust objective problem

```
Input : Maximum number of iterations $M$
Initialize $\mathscr{U}' := \{\xi_0\} \subseteq \mathscr{U}$
for $iter = 1$ to $M$ do
    Set $x^* \in \arg\min_{x \in \mathscr{X}} \max_{\xi \in \mathscr{U}'} c(x,\xi)$
    Set $\xi^* \in \arg\max_{\xi \in \mathscr{U}} c(x^*,\xi)$
    if $c(x^*,\xi^*) > \max_{\xi \in \mathscr{U}'} c(x^*,\xi)$ then
        Add $\xi^*$ to $\mathscr{U}'$
    else
        break
return $x^*$
```

Algorithm 4 : Iterative constraint generation for robust constraint problem

```
Input : Maximum number of iterations $M$
Initialize $\mathscr{U}' := \{\xi_0\} \subseteq \mathscr{U}$
for $\mathtt{iter} = 1$ to $M$ do
    Set $x^* \in \arg\min_{x \in \mathscr{X}: c(x,\xi) \leq 0, \forall \xi \in \mathscr{U}'} f(x, \xi)$
    Set $\xi^* \in \arg\max_{\xi \in \mathscr{U}} c(x^*, \xi)$
    if $c(x^*, \xi^*) > 0$ then
        Add $\xi^*$ to $\mathscr{U}'$
    else
        break
return $x^*$
```

Algorithm for Deep Cluster then Classify with deep K-means

```
Algorithm 5 : Deep Cluster then Classify with deep $K$-means
Input : Dataset $\mathscr{D}_{\xi,\psi}$; number of clusters $K$; maximum number of iterations $T$;
        coverage error $\varepsilon$
Randomly initialize $\theta_0$, $V_0$, and all $W_0^k$
Let $a_0(\psi) := \bar{a}^{\theta_0}(g_{V_{E0}}(\psi))$
Set $t := 0$
repeat
    Set $t := t+1$
    Update $\theta_t^k := \sum_{i\in\mathscr{I}_k} g_{V_{Et-1}}(\psi_i)/|\mathscr{I}_k|$, where $\mathscr{I}_k := \{i : a_{t-1}(\psi_i) = k\}$
    Let $a_t(\psi) := \bar{a}^{\theta_t}(g_{V_{Et-1}}(\psi))$
    Update $V_t$ using SGD on Eq. (1.8) with $a_t(\psi_i)$
until $t \geq T$
Let $a(\psi) := a_t(\psi)$
for $k = 1$ to $K$ do
    Train the parameters $W^k$ using Eq. (1.5) with $\mathscr{D}_\xi^k$
    Calibrate $R^k$ on $\mathscr{D}_\xi^k$ using coverage target $1-\varepsilon$
    Let $\mathscr{U}^k := \mathscr{U}(W^k, R^k, \mathscr{S}^k)$
return $a(\cdot)$ and $\{\mathscr{U}^k\}_{k=1}^K$
```

References

Ban, G.-Y., J. Gallien, and A. J. Mersereau. 2019. “Dynamic procurement of new products with covariate information: The residual tree method.” *Manufacturing & Service Operations Management* 21 (4): 798–815.

Ban, G.-Y., and C. Rudin. 2019. “The big data newsvendor: Practical insights from machine learning.” *Operations Research* 67 (1): 90–108.

Bendsøe, M. P., A. Ben-Tal, and J. Zowe. 1994. “Optimization methods for truss geometry and topology design.” *Structural optimization* 7 (3): 141–159.

Bernardo, F. P., and P. M. Saraiva. 1998. “Robust optimization framework for process parameter and tolerance design.” *AIChE Journal* 44 (9): 2007–2017.

Bertsimas, D., and N. Kallus. 2020. “From predictive to prescriptive analytics.” *Management Science* 66 (3): 1025–1044.

Bertsimas, D., C. McCord, and B. Sturt. 2022. “Dynamic optimization with side information.” *European Journal of Operational Research.*

Bertsimas, D., O. Nohadani, and K. M. Teo. 2007. “Robust optimization in electromagnetic scattering problems.” *Journal of Applied Physics* 101 (7): 074507.

Bertsimas, D., and B. Van Parys. 2021. “Bootstrap robust prescriptive analytics.” *Mathematical Programming,* 1–40.

Chang, J., L. Wang, G. Meng, S. Xiang, and C. Pan. 2017. “Deep adaptive image clustering.” In *Proceedings of the IEEE international conference on computer vision,* 5879–5887.

Chu, M., Y. Zinchenko, S. G. Henderson, and M. B. Sharpe. 2005. “Robust optimization for intensity modulated radiation therapy treatment planning under uncertainty.” *Physics in Medicine and Biology* 50, no. 23 (November): 5463–5477.

Donti, P., B. Amos, and J. Kolter. 2017. “Task-based End-to-end Model Learning.” *CoRR* abs/1703.04529.

Elmachtoub, A. N., and P. Grigas. 2022. “Smart “predict, then optimize”.” *Management Science* 68 (1): 9–26.

Fard, M. M., T. Thonet, and E. Gaussier. 2020. “Deep k-means: Jointly clustering with k-means and learning representations.” *Pattern Recognition Letters* 138:185–192.

Goerigk, M., and J. Kurtz. 2020. “Data-Driven Robust Optimization using Unsupervised Deep Learning.” *arXiv preprint arXiv:2011.09769.*

Guo, X., L. Gao, X. Liu, and J. Yin. 2017. “Improved deep embedded clustering with local structure preservation.” In *Ijcai,* 1753–1759.

Hannah, L., W. Powell, and D. Blei. 2010. “Nonparametric density estimation for stochastic optimization with an observable state variable.” *Advances in Neural Information Processing Systems* 23.

Hu, Y., N. Kallus, and X. Mao. 2022. “Fast rates for contextual linear optimization.” *Management Science.*

Hu, Y., S. Zhang, X. Chen, and N. He. 2020. “Biased stochastic first-order methods for conditional stochastic optimization and applications in meta learning.” *Advances in Neural Information Processing Systems* 33:2759–2770.

Ji, P., T. Zhang, H. Li, M. Salzmann, and I. Reid. 2017. “Deep subspace clustering networks.” *Advances in neural information processing systems* 30.

Kallus, N., and X. Mao. 2020. “Stochastic optimization forests.” *arXiv preprint arXiv:2008.07473.*

Kannan, R., G. Bayraksan, and J. Luedtke. 2021. “Heteroscedasticity-aware residuals-based contextual stochastic optimization.” *arXiv preprint arXiv:2101.03139.*

Kannan, R., G. Bayraksan, and J. R. Luedtke. 2020. "Data-driven sample average approximation with covariate information." *Optimization Online. URL: http://www. optimization-online. org/DB HTML/2020/07/7932. html.*

Lin, S., Y. (Chen, Y. Li, and Z.-J. M. Shen. 2022. "Data-Driven Newsvendor Problems Regularized by a Profit Risk Constraint." *Production and Operations Management* 31 (4): 1630–1644.

Mani, M., A. K. Singh, and M. Orshansky. 2006. "Joint Design-Time and Post-Silicon Minimization of Parametric Yield Loss using Adjustable Robust Optimization." In *2006 IEEE/ACM International Conference on Computer Aided Design,* 19–26.

McCord, C. G. 2019. "Data-driven dynamic optimization with auxiliary covariates." PhD diss., Massachusetts Institute of Technology.

Moradi Fard, M., T. Thonet, and E. Gaussier. 2020. "Deep k-Means: Jointly clustering with k-Means and learning representations." *Pattern Recognition Letters* 138:185–192.

MOSEK ApS. 2022. *MOSEK Fusion API for C++ 9.3.20.* https://docs.mosek.com/latest/cxxfusion/index.html.

Natarajan, K., D. Pachamanova, and M. Sim. 2008. "Incorporating asymmetric distributional information in robust value-at-risk optimization." *Management Science* 54 (3): 573–585.

Nguyen, V. A., F. Zhang, J. Blanchet, E. Delage, and Y. Ye. 2021. *Robustifying conditional portfolio decisions via optimal transport.*

Ohmori, S. 2021. "A Predictive Prescription Using Minimum Volume k-Nearest Neighbor Enclosing Ellipsoid and Robust Optimization." *Mathematics* 9 (2): 119.

Page Jr, T. J. 1984. "Multivariate statistics: A vector space approach." *JMR, Journal of Marketing Research (pre-1986)* 21 (000002): 236.

Rockafellar, R. T., and R. J.-B. Wets. 2009. *Variational analysis.* Vol. 317. Springer Science & Business Media.

Srivastava, P. R., Y. Wang, G. A. Hanasusanto, and C. P. Ho. 2021. "On Data-Driven Prescriptive Analytics with Side Information: A Regularized Nadaraya-Watson Approach." *arXiv preprint arXiv:2110.04855.*

Wang, C., X. Peng, C. Shang, C. Fan, L. Zhao, and W. Zhong. 2021. "A deep learning-based robust optimization approach for refinery planning under uncertainty." *Computers & Chemical Engineering* 155:107495.

Wang, K., and A. Jacquillat. 2020. "From classification to optimization: A scenario-based robust optimization approach." *Available at SSRN 3734002.*

Xu, Y., and S. B. Cohen. 2018. "Stock movement prediction from tweets and historical prices." In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* 1970–1979.

Chapter 2

End-to-End Conditional Robust Optimization

Abstract

The field of Contextual Optimization (CO) integrates machine learning and optimization to solve decision making problems under uncertainty. Recently, a risk sensitive variant of CO, known as Conditional Robust Optimization (CRO), combines uncertainty quantification with robust optimization in order to promote safety and reliability in high stake applications. Exploiting modern differentiable optimization methods, we propose a novel end-to-end approach to train a CRO model that accounts for both the empirical risk of the prescribed decisions and the quality of conditional coverage of the contextual uncertainty set that supports them. While guarantees of success for the latter objective are impossible to obtain from the point of view of conformal prediction theory, high quality conditional coverage is achieved empirically by ingeniously employing a logistic regression differentiable layer within the calculation of coverage quality in our training loss. We show that the proposed training algorithms produce decisions that outperform the traditional "estimate then optimize" approaches.

2.1 Introduction

In a standard machine learning setting, $\Psi \subseteq \mathbb{R}^m$ represent the input set and $\Xi \subseteq \mathbb{R}^m$ represent the output sets and we aim to learn a model $\mathfrak{F}_\theta$ parameterized by θ that approximates the relationship between the input and output by minimizing a loss function $\mathscr{L}$. In real-world applications, we usually have a dataset of M samples, $\mathscr{D}_{\psi\xi} := \{(\psi_i, \xi_i)\}_{i=1}^M$ which are used to approximate the underlying input-output relationship learned by the model. For a new data sample $\psi \in \Psi$, the model trained on $\mathscr{D}_{\psi\xi}$ is used to predict a corresponding target $\xi = \mathfrak{F}_\theta(\psi)$. Recently, there has been a growing interest in developing data-driven optimization solutions that integrate this learning process with the subsequent optimization process. In this context, one accounts for the fact that the prediction is used within a cost minimization problem $\hat{x}^*(\psi) := \arg\min_{x\in\mathscr{X}} c(x, \mathfrak{F}_\theta(\psi))$, where $\mathscr{X} \subseteq \mathbb{R}^n$ is the set of feasible decisions and $c(x,\xi)$ the cost function. The intent is to adapt the training procedure to produce an adapted decision with low out-of-sample expected cost $\mathbb{E}[c(\hat{x}^*(\psi), \xi)]$.

When there is a mismatch between the training loss $\mathscr{L}$ and the cost function $c(x,\xi)$, a small error in predicting ξ for a given ψ can lead to highly suboptimal $x^*(\psi)$ (Elmachtoub and Grigas 2022). Task-based (or decision-focused) learning (Mandi et al. 2024; Donti, Amos, and J. Z. Kolter 2017) addresses this issue by training the model $\mathfrak{F}_\theta$ directly on the performance of the policy $x^*(\psi)$. By trading off predictive performance in favor of task performance, the task-based approach can give near optimal decisions.

In high stakes applications, a Decision Maker (DM) usually demonstrates a certain degree of risk aversion by requiring some level of protection against a range of plausible future scenarios. A natural risk averse variant of integrated learning and optimization takes the form of Conditional Robust Optimization (CRO) (Chenreddy, Bandi, and Delage 2022), which integrates conformal prediction with robust optimization. Specifically, machine learning is first used to estimate an uncertainty set $\mathscr{U}(\psi)$ for an observed context ψ. This set $\mathscr{U}(\psi)$, known to contain the realized ξ with a high probability, is then inserted into the conditional robust optimization model:

$$x^*(\psi) := \arg\min_{x\in\mathscr{X}} \max_{\xi\in\mathscr{U}(\psi)} c(x,\xi) \tag{2.1}$$

To this date, the methods proposed in the conditional robust optimization literature follow an Estimate Then Optimize (ETO) paradigm. Namely, data is first used to estimate the contextual uncertainty sets which are then calibrated to meet the required coverage levels. These sets are then used as input to the CRO problem to get the adapted robust decision $x^*(\psi)$. However, the process of calibrating uncertainty sets does not take into account the downstream optimization task, potentially resulting in misalignment between the loss function used in the initial estimation and the objective of robust optimization.

In this chapter, we propose a novel end-to-end learning framework for conditional robust optimization that constructs the contextual uncertainty set by accounting for the downstream task loss. Our contributions can be described as follows:

- We propose for the first time an end-to-end training algorithm to produce contextual uncertainty sets, $\mathscr{U}(\psi)$ that lead to reduced risk exposure for the solution of the down-stream CRO problem
- We introduce a novel joint loss function aimed at enhancing the conditional coverage of $\mathscr{U}(\psi)$ while improving the CRO objective
- We demonstrate through a set of synthetic environments that our end-to-end approach surpasses ETO approaches at the CRO task while achieving comparable if not superior conditional coverage with its learned contextual set
- We show empirically how our end-to-end learning approach outperforms other state-of-the-art methods on a portfolio optimization problem using real world data from the U.S. stock market

Remark 2.1.1. *It is worth noting that when the estimated uncertainty set $\mathscr{U}(\psi)$ reduces to a singleton $\{\mathfrak{F}_\theta(\psi)\}$, i.e. a point prediction, the CRO problem simplifies to the deterministic contextual optimization problem: $x^*(\psi) := \arg\min_{x\in\mathscr{X}} c(x,\mathfrak{F}_\theta(\psi))$. For this* *special case, the training of $\mathfrak{F}_\theta(\psi)$ using an end-to-end paradigm has been more heavily studied, see for instance Amos and Kolter (2017), Berthet et al. (2020), and Elmachtoub and Grigas (2022). End-to-end CRO therefore constitutes a more general and unexplored framework that can potentially answer to a need to provide more robust decisions in situations where parameters cannot be perfectly estimated. This is particularly noticeable in a portfolio optimization problem where a point estimate of the return of assets will necessarily motivate investing all available wealth in the one single asset with highest predicted return. In contrast, it is rather easy to formulate an uncertainty set $\mathscr{U}(\psi)$ such that the CRO problem encourage diversification of the investment.*

2.2 Related Work

Estimate Then Optimize popularized by the pioneering work of Hannah, Powell, and Blei (2010) is a framework that integrates machine learning and optimization tasks. Several approaches are proposed to learn the conditional distribution from data. Kannan, Bayraksan, and J. R. Luedtke (2020) and Sen and Deng (2018) propose using residuals from the trained regression model to learn conditional distributions. Bertsimas and Kallus (2020) assign weights to the historical observations of the parameters and solve the weighted SAA problem. We refer the readers to the Mišić and Perakis (2020) survey for various applications of the ETO framework. Besides the mentioned risk neutral applications, there is a growing interest in integrating machine learning techniques to Robust Optimization to handle risk-averse scenarios. Chenreddy, Bandi, and Delage (2022) identify clusters of the uncertain parameters based on the covariate data and calibrate the sets for these clusters. Y. Patel, Rayan, and Tewari (2023) propose using non-convex prediction regions to construct uncertainty sets. Blanquero, Carrizosa, and Gómez-Vargas (2023) construct contextual ellipsoidal uncertainty sets by making normality assumptions. Ohmori (2021) use a non-parametric K-nearest neighbors model to identify the minimum volume ellipsoid to be used as an uncertainty set. Sun, Liu, and Li (2023) solve a robust contextual LP problem where a prediction model is first learned, and then uncertainty is calibrated to match robust objectives. It is to be noted that all these CRO approaches follow the ETO paradigm.

End-to-end learning is a more recent stream of work that integrates the Estimation and Optimization tasks and trains using the downstream loss. Donti, Amos, and J. Z. Kolter (2017) proposed using an end-to-end approach for learning probabilistic machine learning models using task loss. Elmachtoub and Grigas (2022) learn contextual point predictor by minimizing the regret associated with implementing prescribed action based on such a point predictor. Amos and Kolter (2017) use implicit differentiation methods to train an end-to-end model. Butler and Kwon (2023) solve large-scale QPs using the ADMM algorithm that decouples the differentiation procedure for primal and dual variables. Elmachtoub and Grigas (2022) and Mandi, Stuckey, Guns, et al. (2020) propose using a surrogate loss function to train integrated methods to address loss functions with non-informative gradients. I. Wang et al. (2023) propose learning a non-contextual uncertainty set by maximizing the expected performance across a set of randomly drawn parameterized robust constrained problems while ensuring guarantees on the probability of constraint satisfaction with respect to the joint distribution over perturbance and robust problems. Costa and Iyengar (2023) propose a distributionally robust end-to-end system that integrates residual based distribution estimation and robustness tuning to the portfolio construction problem. We refer the reader to Kotary et al. (2021), Qi and Shen (2022), Mandi et al. (2024), and Sadana et al. (2025) for broader discussions on both ETO and end-to-end approaches.

Uncertainty quantification methods are employed to estimate the confidence of deep neural networks over their predictions (Kontolati et al. 2022). Common uncertainty quantification approaches include using Bayesian methods like stochastic deep neural networks, ensembling over predictions from several models to suggest intervals, and models that directly predict uncertain intervals (Gawlikowski et al. 2021). Beyond estimating predictive uncertainty, ensuring its statistical reliability is crucial for safe decision-making (C. Guo et al. 2017). Conformal prediction has become popular as a distribution-free calibration method (Shafer and Vovk 2008). Although conformal prediction ensures marginal coverage, attaining conditional coverage in the most general case is desirable (Vovk 2012). Although considered infeasible, Romano et al. (2020) offers group conditional guarantees for disjoint groups by independently calibrating each group.

2.3 Estimate then Robust Optimize

The concept of "Estimate Then Optimize" comes from the contextual optimization literature, as discussed by Sadana et al. (2025). In the context of CRO, the role of the **Estimation** process is to quantify the uncertainty about ξ given the observed ψ. This is given as input to an **Optimization** problem that prescribes an optimal contextual decision $x^*(\psi)$.

When the downstream optimization problem is a CRO problem, the estimation step is required to produce a region that adapts to the observed covariates ψ and is expected to contain the response ξ with high confidence. This can be executed in two steps: first, by learning a parametric conditional distributional model denoted as $F_\theta(\psi)$, and second, by calibrating an implied confidence region $\mathscr{U}_\theta(\psi)$ to ensure $\mathbb{P}_{F_\theta(\psi)}(\xi \in \mathscr{U}_\theta(\psi)) = 1-\varepsilon$. For e.g., when one assumes that $\xi|\psi \sim \mathscr{N}(\hat{\mu}(\psi), \hat{\Sigma}(\psi))$, one can learn $(\hat{\mu}(\psi), \hat{\Sigma}(\psi))$ by maximizing the log-likelihood function (see Barratt and Boyd (2023))

$$-\frac{n}{2}\log(2\pi) + \sum_{j=1}^{n} \log L(\psi)_{jj} - \frac{1}{2}\|L(\psi)^\top(\xi - \hat{v}(\psi))\|_2^2$$

where $L(\psi)$ and $\hat{v}(\psi)$ are the parametric mappings that can be used to compose $\hat{\mu}(\psi) := (L(\psi)^{-1})^\top v(\psi)$ changed from $L(\psi)^\top$ to $(L(\psi)^{-1})^\top$ and $\hat{\Sigma}(\psi) = (L(\psi)^{-1})^\top L(\psi)^{-1}$. Using the α quantile from the chi-squared distribution with m degrees of freedom, one can define $\mathscr{U}_\theta(\psi)$ that satisfies $\mathbb{P}(\xi \in \mathscr{U}_\theta(\psi)) = 1-\varepsilon$ asymptotically.

Some recent work completely circumvents the need for the intermediary F_θ by calibrating some $\mathscr{U}_\theta(\psi)$ directly on the dataset. For example, Chenreddy, Bandi, and Delage (2022) propose identifying a k-class classifier, $a : \mathbb{R}^m \to [K]$ to reduce $\mathscr{U}_\theta(\psi) := \mathscr{U}_\theta(a(\psi))$ such that $\mathbb{P}(\xi \in \mathscr{U}_\theta(k)|a(\psi) = k) \geq 1-\varepsilon \ \forall\, k$. The literature on conformal prediction also belongs to the family of distribution-free approaches. It separates the calibration of the shape of $\mathscr{U}_\theta(\psi)$ from the calibration of its size, parameterized by a radius $r > 0$, on a reserved validation set to provide out-of-sample marginal coverage guarantees of the form $\mathbb{P}(\xi \in \mathscr{U}_\theta(\psi)) \geq 1-\varepsilon$, where the probability is taken over both the draw of the validation set and of the next sample.According to the Lemma 4.2 in Chenreddy, Bandi, and Delage (2022), such a coverage guarantee is sufficient to ensure that the out-of-sample Value-at-risk of the robust policy produced by CRO is bounded above by the worst-case value of the in-sample problem.

2.4 End-to-End Conditional Robust Optimization

While the ETO approach presented in Section 2.3 presents an efficient way to quantify the uncertainty conditionally, it does not take into account the quality of decisions $x^*(\psi)$ prescribed by the downstream CRO model. In practice, the quality of a robust decision is usually assessed by measuring the risk associated with the cost produced on a new data sample (a.k.a. out-of-sample). We assume that this risk is measured by a risk measure that reflects the amount of risk aversion experienced by the DM. For instance, one can use conditional value-at-risk represented by the function, $\rho_\alpha(X) := \inf_t t + (1/(1-\alpha))\mathbb{E}[(X-t)^+]$, which computes the expected value in the right tail of the random cost X for a certain risk aversion α and it covers both expected value and the worst-case cost as special cases (i.e. when $\alpha = 0$ and 1 respectively). In an ETO framework, once the optimal decision $x^*(\psi)$ is determined, the DM can assess the associated risk, also known as task loss, $\rho_\alpha(c(x^*(\psi),\xi))$. This metric allows comparison across models to select the suitable one. However, it is important to note that the model with the best performance in terms of task loss may differ from the optimal model based on prediction loss. Motivated by recent evidence from Elmachtoub and Grigas (2022) indicating that performance improvement can be achieved by employing a decision-focused/ task-based learning paradigm, we propose end-to-end conditional robust optimization.

2.4.1 The ECRO Training Problem

Formally, we let $\Psi \subseteq \mathbb{R}^m$ be an arbitrary support set for ψ whereas $\Xi \subseteq \mathbb{R}^m$ is assumed for simplicity to be contained within a ball centered at 0 of radius R_ξ. We consider $c(x, \xi)$ to be convex in x and concave in ξ and let $\mathscr{X}(\psi) := \{x \in \mathbb{R}^n | g(x, \psi) \leq 0, h(x, \psi) = 0\}$ be a convex feasible set for x, possibly dependent on ψ, and defined through a set of convex inequalities, identified using $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^J$ and affine equalities, identified using an affine mapping $h : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^J$. The conditional optimal policy then becomes:

$$x^*(\psi, \mathscr{U}) := \arg \min_{x \in \mathscr{X}(\psi)} \max_{\xi \in \mathscr{U}(\psi)} c(x, \xi), \tag{2.2}$$

where we make explicit how the decision depends on both the contextual uncertainty set and the realized covariate. Given a parametric family of contextual uncertainty set $\mathscr{U}_\theta$ with $\theta \in \Theta$ and a dataset $D_{\psi\xi} := \{(\psi^i, \xi^i)\}_{i=1}^M$, the ECRO training problem consists in identifying

$$\min_{\theta \in \Theta} \mathscr{L}_{ECRO}(\theta) := \rho_{i \sim M}(c(x^*(\psi^i, \mathscr{U}_\theta), \xi^i)), \tag{2.3}$$

where $\rho_{i \sim M}$ refers to the risk when i is drawn uniformly from 1 to M, while, for simplicity, we assume $\rho(\cdot)$ to be a conditional value-at-risk measure, and $\mathscr{U}_\theta(\psi)$ to be ellipsoidal for all ψ. Namely, we can assume that

$$\begin{aligned} \mathscr{U}_\theta(\psi) &= \mathscr{E}(\mu_\theta(\psi), \Sigma_\theta(\psi), r) \\ &:= \{\ \xi \in \mathbb{R}^m : (\xi - \mu_\theta(\psi))^T \Sigma_\theta(\psi)^{-1} (\xi - \mu_\theta(\psi)) \leq 1\}, \end{aligned} \tag{2.4}$$

for some $\mu_\theta : \mathbb{R}^m \to \mathbb{R}^m$ and $\Sigma_\theta : \mathbb{R}^m \to \mathscr{S}_+$, where $\mathscr{S}_+$ is the set of positive definite matrices, for all $\theta \in \Theta$. While the robust optimization literature suggests various uncertainty set structures that facilitate the resolution of the RO problem, the ellipsoidal set stands out as a natural one to employ as it retains numerical tractability (see Ben-Tal and Nemirovski (1998)) and can easily be described to the DM.

The training pipeline for the task-based learning approach is illustrated in Figure 2.1. In this pipeline, one starts from an arbitrary θ^0, the optimization problem (2.2) is solved

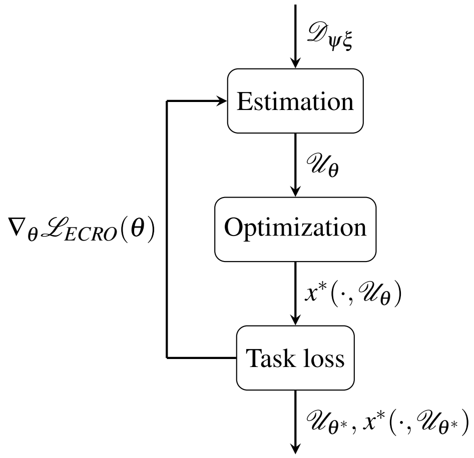

$\mathscr{D}_{\psi\xi}$
Estimation
$\mathscr{U}_\theta$
Optimization
$x^*(\cdot, \mathscr{U}_\theta)$
Task loss
$\mathscr{U}_{\theta^*}, x^*(\cdot, \mathscr{U}_{\theta^*})$
$\nabla_\theta \mathscr{L}_{ECRO}(\theta)$

Figure 2.1: Training pipeline for task-based learning

first for each data point, and the resulting optimal actions are then implemented in order to measure the empirical risk under $D_{\psi\xi}$, which we call empirical ECRO loss of θ^0. A gradient of $\mathscr{L}_{ECRO}(\theta)$ can then be used to update θ^0 in a direction of improvement. Key steps in this pipeline consist of computing $x^*(\psi^i, \mathscr{U}_\theta)$ efficiently and in a way that enables differentiation with respect to θ.

2.4.2 Reducing and Solving the Robust Optimization Task

Given the convex-concave structure of $c(x,\xi)$ and the convexity and compactness of the ellipsoidal set, we can employ Fenchel duality (see Ben-Tal, Den Hertog, and Vial 2015) to reformulate the min-max problem as a simpler minimization form over an augmented decision space. Specifically, we first replace the original cost function with the equivalent cost

$$\bar{c}(x,\xi) := \begin{cases} c(x,\xi) & \text{if } \|\xi\|_2 \leq R_\xi \\ -\infty & \text{otherwise} \end{cases},$$

which integrates information about the domain of ξ. One can then employ theorem 6.2 of Ben-Tal, Den Hertog, and Vial (2015), to show that problem (2.1) can be reformulated as:

$$\min_{x \in \mathscr{X}(\psi), v} f(x,v,\psi) := \delta^*(v|\mathscr{U}_\theta(\psi)) - \bar{c}_*(x,v) \tag{2.5}$$

where the support function

$$\delta^*(v|\mathscr{U}_\theta(\psi)) := \sup_{\xi \in \mathscr{U}_\theta(\psi)} \xi^T v = \mu_\theta(\psi)^T v + \sqrt{v^T \Sigma_\theta(\psi) v}, \tag{2.6}$$

while the partial concave conjugate function is defined as

$$\bar{c}_*(x,v) := \inf_\xi v^T \xi - \bar{c}(x,\xi) = \inf_{\xi : \|\xi\|_2 \leq R_\xi} v^T \xi - c(x,\xi).$$

This leads to $x^*(\psi, \mathscr{U}(\psi))$ being the minimizer of the convex minimization problem:

$$\min_{x \in \mathscr{X}(\psi), v} f(x,v,\psi) \tag{2.7}$$

with $f(x,v,\psi) := \mu_\theta(\psi)^T v + \sqrt{v^T \Sigma_\theta(\psi) v} - \bar{c}_*(x,v)$, a jointly convex function of x and v and finite valued over its domain, and with sub-derivatives:

$$\begin{aligned} \nabla_v f(x,v,\psi) &= \mu_\theta(\psi) + (1/\sqrt{v^T \Sigma_\theta(\psi) v}) \Sigma_\theta(\psi) v - \xi^*(x,v) \\ \nabla_x f(x,v,\psi) &= \nabla_x c(x, \xi^*(x,v)), \end{aligned}$$

where $\xi^*(x,v) := \text{argmin}_{\xi : \|\xi\|_2 \leq R_\xi} v^T \xi - c(x,\xi)$. Revisiting the procedure outlined in Figure 2.1, one can observe that the training process requires a forward pass to find the optimal solutions and a backward pass to update the parameter vector θ. This requires the computation of the gradients of the solution to the problem (2.3) with respect to the input parameters that are passed through the reformulated CRO problem. Furthermore, the minimization procedure in problem (2.3) entails navigating through the risk measure ρ. These aspects will be further explored in the next section.

2.4.3 Gradient for Problem (2.3)

In training problem (2.3), the gradient of $\mathscr{L}_{ECRO}(\theta)$ with respect to θ can be obtained using the chain rule:

$$\begin{aligned}\nabla_\theta \mathscr{L}_{ECRO}(\theta) = \sum_i \frac{\partial \rho_{i\sim M}(y_i)}{\partial y_i}\big|_{y_i=c(x^*(\psi^i,\mathscr{U}_\theta),\xi^i)} \cdot \\ \nabla_x c(\boldsymbol{x})\big|_{x=x^*(\psi^i,\mathscr{U}_\theta)} \cdot \\ \Big(\nabla_\mu x^*(\psi^i,\mathscr{E}(\mu,\Sigma_\theta(\psi^i)))\big|_{\mu=\mu_\theta(\psi^i)}\nabla_\theta\mu_\theta(\psi^i) \\ +\nabla_\Sigma x^*(\psi^i,\mathscr{E}(\mu_\theta(\psi^i),\Sigma))\big|_{\Sigma=\Sigma_\theta(\psi^i)}\nabla_\theta\Sigma_\theta(\psi^i)\Big)\end{aligned}$$

Based on Ruszczyński and Shapiro (2021), when $\rho(Y) := \text{CVaR}_\alpha(Y)$, one can employ the subdifferential:

$$\nabla_{\boldsymbol{y}}\rho_{i\sim M}(y_i) = \boldsymbol{\upsilon}(\boldsymbol{y})$$

with $\boldsymbol{\upsilon}(\boldsymbol{y}) \in \text{argmax}_{\boldsymbol{\upsilon}\in\mathbb{R}^M_+:\mathbf{1}^T\boldsymbol{\upsilon}=1,\boldsymbol{\upsilon}\leq((1-\alpha)N)^{-1}} \boldsymbol{\upsilon}^T\boldsymbol{y}$.

Given that $\nabla_{\boldsymbol{x}}c(\boldsymbol{x})$, $\nabla_\theta\mu_\theta(\psi)$, and $\nabla_\theta\Sigma_\theta(\psi)$ can be readily obtained using auto-differentiation (Seeger et al. 2017) when $c(\boldsymbol{x})$, $\mu_\theta(\psi)$, and $\Sigma_\theta(\psi)$ are differentiable, we focus the rest of this subsection on the process of identifying $\nabla_{(\mu,\Sigma)}x^*(\psi,\mathscr{E}(\mu,\Sigma))$. Following the decision-focus learning literature (see Blondel et al. 2022), one can identify such derivatives by exploiting the fact that any optimal primal-dual pair $(x^*,v^*,\lambda^*,\nu^*)$ of problem (2.7) must satisfy the Karush-Kuhn-Tucker (KKT) conditions, which take the form:

$$G(x^*,v^*,\lambda^*,\nu^*,\mu,\Sigma,\psi) = 0, \qquad g(x^*,\psi) \leq 0, \lambda^* \geq 0.$$

where

$$G(x^*,v^*,\lambda^*,\nu^*,\mu,\Sigma,\psi) := \begin{bmatrix}\nabla_x f(x^*,v^*,\psi) + \nabla_x g(x^*,\psi)^T\lambda^* + \nabla_x h(x^*,\psi)^T\nu^* \\ \lambda^*\circ g(x^*,\psi) \\ h(x^*,\psi)\end{bmatrix}$$

and $\circ$ denotes the Hadamard product of two vectors.

One can therefore apply implicit differentiation to the constraints $G(x^*, v^*, \lambda^*, \nu^*, \mu, \Sigma, \psi) = 0$ to identify $\nabla_{(\mu,\Sigma)} x^*(\psi, \mathscr{E}(\mu,\Sigma))$ simultaneously with the derivatives of v^*, λ^*, and ν^* with respect to the pair (μ,Σ). Specifically, one is required to solve the system of equations:

$$\begin{aligned}&\frac{\partial}{\partial x, v, \lambda, \nu} G(x^*, v^*, \lambda^*, \nu^*, \mu, \Sigma, \psi)\cdot\\&\frac{\partial}{\partial(\mu,\Sigma)}(x^*, v^*, \lambda^*, \nu^*)(\mu,\Sigma) =\\&-\frac{\partial}{\partial(\mu,\Sigma)} G(x^*, v^*, \lambda^*, \nu^*, \mu, \Sigma, \psi),\end{aligned}$$

where $\frac{\partial}{\partial(x,v,\lambda,\nu)}G$ denotes the Jacobian of the mapping G with respect to (x, v, λ, ν). We refer to Blondel et al. (2022) and Duvenaud, Kolter, and Johnson (2020) for further details on the computations of related to implicit differentiation.

2.4.4 Task-based Set (TbS) Algorithm

In this section, we delve into the implementation details of the ECRO training pipeline. Regarding the contextual ellipsoidal set $\mathscr{E}(\mu_\theta(\psi), \Sigma_\theta(\psi))$, we follow the ideas proposed in Barratt and Boyd (2023) and employ a neural network that maps from $\mathfrak{F}_\theta : \mathbb{R}^m \to \mathbb{R}^m \times \mathbb{R}^{m(m+1)/2} \times \mathbb{R}$. The first set of outputs is used to define $\mu_\theta(\psi)$ while the second and third set forms a lower triangular matrix $L_\theta(\psi)$ and scalar $r_\theta(\psi)$, which is made independent of ψ w.l.o.g., used to produce $\Sigma_\theta(\psi) := r_\theta(\psi) L_\theta(\psi) L_\theta(\psi)^T$. The positive definiteness of $\Sigma_\theta(\psi)$ is ensured by taking an exponential in the last layer of the network for the output that appears in the diagonal of L. The architecture of the neural network can be found in Appendix 2.8.2.

The second set of notable details has to do with solving for $x^*(\psi^i, \mathscr{E}(\mu_\theta^i, \Sigma_\theta^i, r_\theta))\ \forall i$. In our implementation of end-to-end learning for conditional robust optimization, we found that a trust region optimization (TRO) method (see Byrd, Gilbert, and Nocedal 2000) could efficiently solve the reformulated robust optimization problem (2.7) and provide primal-dual solution pairs for this problem. Given that each episode of the training would pass through the same set of data points, we further observed that the training accelerated significantly (see Figure 2.6 in Appendix 2.8.2) when the trust region was interrupted early (after $K = 5$ iterations) as long as it would be warm started at the solution found at the previous epochs. Algorithm 6 presents our proposed training framework for the ECRO approach.

Algorithm 6 : ECRO Training with Trust Region Solver

```
Input : Dataset $\mathscr{D}_{\xi,\psi}$; max epochs $T$; max TRO steps $K$; batch size $N$; protection
        level $\alpha$
Initialize warm start buffer $\{\bar{x}_1, \dots, \bar{x}_M\}$ with each $\bar{x}^i \in \mathscr{X}(\psi_i)$
Initialize network parameters $\theta$ and set $t := 1$
while not converged and $t \leq T$ do
    Sample a batch of $N$ indices $\mathscr{B} \subset \{1, \dots, M\}$
    for $i \in \mathscr{B}$ do
                                                        // Run TRO for up to $K$ steps
        $(x_i^t, \lambda_i^t, \nu_i^t) \leftarrow \mathrm{TRO}(\bar{x}_i, \mu_\theta(\psi_i), \Sigma_\theta(\psi_i), K)$
        $\bar{x}_i \leftarrow x_i^t$                                   // Update warm start
    Compute $\mathscr{L}_{\mathrm{ECRO}}(\theta)$ and $\nabla_\theta \mathscr{L}_{\mathrm{ECRO}}(\theta)$ for $i \sim \mathscr{B}$
    $\theta \leftarrow \theta - \text{step size} \cdot \nabla_\theta \mathscr{L}_{\mathrm{ECRO}}(\theta)$
    $t \leftarrow t + 1$
return $\theta$
```

2.5 End-to-End CRO with Conditional Coverage

Recall that the ETO framework summarized in Section 2.3 focused on producing contextual uncertainty set with appropriate marginal coverage (of $1 - \varepsilon$) of the realization of ξ. The training pipeline in Section 2.4 was at the other end of the spectrum, disregarding entirely the objective of coverage to increase task performance. In practice, coverage can be a heavy price to pay to obtain performance as it implies a loss in the explainability of the prescribed robust decision. It is becoming apparent that many DMs suffer from algorithm aversion (see Burton, Stein, and Jensen 2020) and could be reluctant to implement a robust decision produced from an ill covering uncertainty set.

We further argue that traditional ETO might already face resistance to adoption given the type of coverage property attributed to the ETO sets, i.e. $\mathbb{P}(\xi \in \mathscr{U}(\psi)) = 1-\varepsilon$. Indeed, marginal coverage guarantees only hold in terms of the joint sampling of ψ and ξ. This implies that it offers no guarantees regarding the coverage of ξ given the observed ψ for which the decision is made. In fact, a 90% marginal coverage can trivially be achieved if $\mathscr{U}(\psi)$ returns Ξ when $\psi \in \Psi$, for some arbitrary set Ψ, and otherwise returns $\emptyset$, as long as $\mathbb{P}(\psi \in \Psi) = 1-\varepsilon$. This is clearly an issue for applications with critical safety considerations and motivates seeking conditional coverage in addition to the marginal coverage when designing $\mathscr{U}(\psi)$. In this section, we outline a training procedure that integrates a sub-procedure that enhances the conditional coverage performance.

2.5.1 The Conditional Coverage Training Problem

We start by briefly formalizing the difference between the two types of coverage in the definition below.

Definition 2.5.1. *Given a confidence level $1-\varepsilon$, a contextual uncertainty set mapping $\mathscr{U}(\cdot)$ is said to satisfy* ***marginal coverage*** *if $\mathbb{P}(\xi \in \mathscr{U}(\psi)) = 1-\varepsilon$, and to satisfy* ***conditional coverage*** *if $\mathbb{P}(\xi \in \mathscr{U}(\psi)|\psi) = 1-\varepsilon$ almost surely.*

The following lemma identifies a necessary and sufficient condition for a contextual set to satisfy conditional coverage.

Lemma 2.5.1. *A contextual uncertainty set $\mathscr{U}(\psi)$ satisfies conditional coverage, at confidence $1-\varepsilon$, if and only if*

$$\mathscr{L}_{CC}(\theta) := \mathbb{E}[\,(\mathbb{P}(\xi \in \mathscr{U}(\psi)|\psi) - (1-\varepsilon))^2\,] = 0$$

Proof. For any random variable X, one can show that :

$$\begin{aligned} & X = 1-\varepsilon \text{ a.s} \\ & \quad \Rightarrow \mathbb{E}[(X-(1-\varepsilon))^2] = 1\cdot(1-\varepsilon-(1-\varepsilon))^2 = 0 \end{aligned}$$

and that, since $y^2 \leq 0 \Leftrightarrow y = 0$,

$$\begin{aligned} &\mathbb{E}[(X-(1-\varepsilon))^2] = 0 \\ &\quad \Rightarrow\ (X-(1-\varepsilon))^2 = 0 \text{ a.s. } \Rightarrow\ X = 1-\varepsilon \text{ a.s..} \end{aligned}$$

By letting $X := \mathbb{P}(\xi \in \mathscr{U}_\theta(\psi)|\psi)$, we obtain our result. □

Equipped with Lemma 2.5.1, we formulate the "theoretical" conditional coverage training problem as $\min_{\theta\in\Theta}\ \mathscr{L}_{CC}(\theta)$. Since the true conditional distribution $\mathbb{P}(\xi \in \mathscr{U}_\theta(\psi)|\psi)$ is typically inaccessible to the DM, we propose an approximation that will make $\mathscr{L}_{CC}(\theta)$ practical.

2.5.2 Regression-Based Conditional Coverage Loss

Given a set $\mathscr{U}$, one can define a binary random variable $y(\psi,\xi,\mathscr{U}) := \mathbf{1}\{\xi \in \mathscr{U}(\psi)\}$, and rewrite the conditional probability distribution $\mathbb{P}(\xi \in \mathscr{U}(\psi)|\psi)$ as $\mathbb{P}(y(\psi,\xi,\mathscr{U}) = 1|\psi)$. Using the i.i.d sample data in $\mathscr{D}_{\psi\xi}$, one can approximate this conditional probability using a parametric model, i.e. $\mathbb{P}(y(\psi,\xi,\mathscr{U}){=}\,1|\psi) \approx g_\phi(\psi)$ for some $\phi \in \Phi$. The parameters ϕ can be calibrated by minimizing the negative conditional log-likelihood of $\{y(\psi^i,\xi^i,\mathscr{U})\}_{i=1}^M$:

$$\phi^*(\mathscr{U}) := \arg\min_{\phi\in\Phi} -\frac{1}{M}\sum_{i=1}^{M} \log g_\phi(\psi^i)^{y^i}(1-g_\phi(\psi^i))^{1-y^i}, \tag{2.8}$$

where $y_i := y(\psi^i,\xi^i,\mathscr{U})$. Using the parametric approximation $g_{\phi^*(\mathscr{U})}(\psi) \approx \mathbb{P}(\xi \in \mathscr{U}(\psi)|\psi)$ and replacing the unknown true distribution of (ψ,ξ) with the empirical one, we obtain our regression-based conditional coverage loss function

$$\hat{\mathscr{L}}_{CC}(\theta) := \mathbb{E}^{\mathscr{D}_{\psi\xi}}[(g_{\phi^*(\mathscr{U}_\theta)}(\psi) - (1-\varepsilon))^2].$$

The gradient of $\hat{\mathscr{L}}_{CC}(\theta)$ can be obtained using similar decision-focused training methods as employed for $\mathscr{L}_{ECRO}(\theta)$ given that:

$$\begin{aligned}\nabla_\theta \hat{\mathscr{L}}_{CC} = & \sum_{i=1}^{M} 2(g_{\phi^*(\mathscr{U}_\theta)}(\psi^i) - (1-\varepsilon))\nabla_\phi g_{\phi^*(\mathscr{U}_\theta)}(\psi^i) \cdot \\ & \sum_{j=1}^{M} \partial\phi^*(\mathscr{E}(\mu,\Sigma_\theta(\psi^i)))/\partial y^j \cdot \\ & \Big(\nabla_\mu y^j(\psi^j,\xi^j,\mathscr{E}(\mu,\Sigma_\theta(\psi^j)))\big|_{\mu=\mu_\theta(\psi^j)}\nabla_\theta\mu_\theta(\psi^j) \\ & + \nabla_\Sigma y^j(\psi^j,\xi^j,\mathscr{E}(\mu_\theta(\psi^j),\Sigma))\big|_{\Sigma=\Sigma_\theta(\psi^j)}\nabla_\theta\Sigma_\theta(\psi^j)\Big),\end{aligned}$$

where the main challenges reside again in the step of differentiating through the minimizer of problem (2.8).

2.5.3 Dual Task Based Set (DTbS) Algorithm

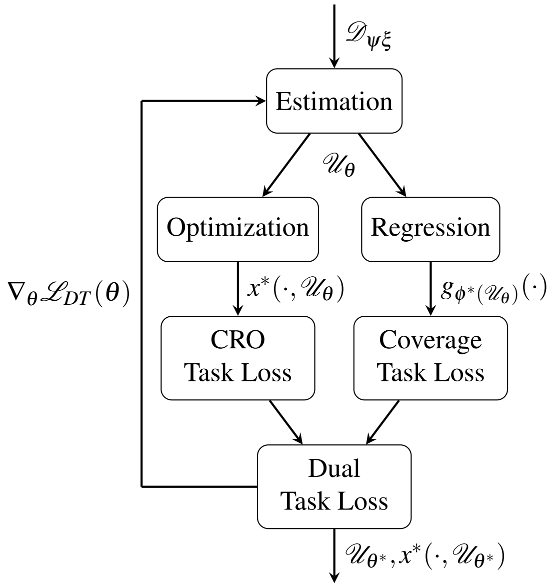

$\mathscr{D}_{\psi\xi}$
Estimation
$\mathscr{U}_\theta$
Optimization
Regression
$\nabla_\theta \mathscr{L}_{DT}(\theta)$
$x^*(\cdot,\mathscr{U}_\theta)$
$g_{\phi^*(\mathscr{U}_\theta)}(\cdot)$
CRO Task Loss
Coverage Task Loss
Dual Task Loss
$\mathscr{U}_{\theta^*}, x^*(\cdot,\mathscr{U}_{\theta^*})$

Figure 2.2: Training pipeline for dual task based learning

We conclude this section with the presentation of our novel integrated algorithm that learns the contextual uncertainty set network $\mathfrak{F}_\theta$ by incorporating both the risk mitigation and conditional coverage tasks in the training. Indeed our DTbS training algorithm minimizes the following double task loss function that trades off between the two task objectives:

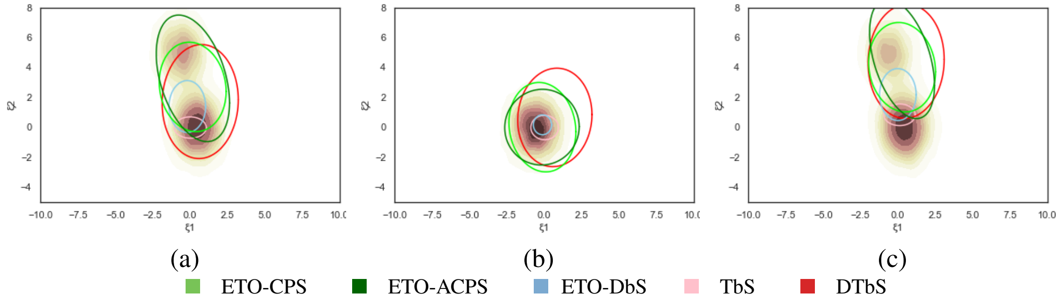
$$\mathscr{L}_{DT}(\theta) = \gamma \mathscr{L}_{ECRO}(\theta) + (1-\gamma)\hat{\mathscr{L}}_{CC}(\theta). \tag{2.9}$$

The training pipeline for this algorithm can be seen in Figure 2.2. It closely mirrors the structure of the TbS algorithm, with additional crucial steps to compute the necessary components of the loss presented in equation (2.9). Within each epoch, the predicted uncertainty set $\mathscr{U}_\theta$ serves two purposes: i) Optimizing CRO to find the optimal policy $x^*(\cdot, \mathscr{U}_\theta)$ and assessing its associated risk; and ii) producing the binary variable $y(\psi, \xi, \mathscr{U}_\theta)$, which regression leading to $g_{\phi^*(\mathscr{U}_\theta)}(\cdot)$ serves to quantify the quality of the conditional coverage. The sum of task losses produces $\mathscr{L}_{DT}(\theta)$, which can be differentiated using decision-focused learning methods. The regression model $g_\phi(\psi)$ takes the form of a feed-forward neural network with a sigmoid activation in the final layer and is optimized using stochastic gradient descent. Algorithm 7 in Appendix 2.8.1 presents the details of this DTbS algorithm.

Remark 2.5.1. *It is to be noted that achieving distribution-free finite sample conditional coverage guarantees is known to be impossible in the conformal prediction literature (see Barber et al. 2020). Recently, some progress has been made towards partial forms of conditional coverage guarantees (see Gibbs, Cherian, and Candès 2023) yet it is unclear what are the implications of exploiting such partial coverage properties for the downstream CRO decisions. It is also unclear how such conditional conformal prediction procedures could be integrated within an end-to-end CRO approach.*

2.6 Experiments

This section outlines our experimental framework devised to demonstrate the advantages of the ECRO method in learning the uncertainty sets tailored to covariate information. Our focus lies in assessing the utility of the model in i) improving the CRO performance; and ii) achieving conditional coverage. We conduct a comparative analysis between our



(a) (b) (c)

Figure 2.3: Comparison of uncertainty set ($\alpha = 0.9$) coverage for different ψ realizations: (a) $[2.5, -0.2]^T$, (b) $[-2.6, 0.5]^T$, (c) $[2.7, 1.9]^T$. The shade indicate the true conditional distribution.

two end-to-end approaches, TbS and DTbS, and three state-of-the-art ETO approaches to formulate contextual ellipsoidal sets. We first consider a Distribution-based contextual ellipsoidal uncertainty Set (ETO-DbS) recently introduced in Blanquero, Carrizosa, and Gómez-Vargas (2023), where the conditional distribution of ξ given ψ is presumed to follow a multivariate normal distribution. Additionally, we explore two distributional-free approaches. A vanilla Conformal Prediction Set (ETO-CPS) uses conformal prediction on the output of a point predictor for ξ given ψ, after shaping the ellipsoid (through an invariant Σ) using the residual errors (see Johnstone and Cox 2021). An Adapted version of Conformal Prediction Set (ETO-ACPS) proposed in Messoudi, Destercke, and Rousseau (2022) adapts the shape Σ using local averaging around the observed ψ. The code can be found on the github[1] repository.

2.6.1 The Portfolio Optimization Application

We explore the effectiveness of the proposed methodologies in addressing a classic robust portfolio optimization problem. In this context, we define the cost function $c(x, \xi)$ as $-\xi^T x$, where x represents a portfolio comprising investments in m different assets, with their respective returns denoted in the random vector ξ. Additionally, we impose constraints on x, encapsulated within $\mathscr{X}$, defined as $\mathscr{X} := \{x \in \mathbb{R}^m | \sum_{i=1}^m x_i = 1, x \geq 0\}$. For this cost

[1] https://github.com/Achenred/End-to-end-CRO

function, we obtain the partial concave conjugate function:

$$\bar{c}_*(x,v) = \inf_{\xi:\|\xi\|_2 \leq R_\xi} v^T\xi - \xi^T x = -R_\xi \|v-x\|_2 \tag{2.10}$$

Thus leading to problem (2.7) becoming

$$\min_{x\in\mathscr{X}} f(x,\psi) := x^T\mu_\theta(\psi) + \sqrt{x^T\Sigma_\theta(\psi)x} \tag{2.11}$$

when $R_\xi \to \infty$, thus capturing $\Xi := \mathbb{R}^m$.

2.6.2 CRO Performance Using Synthetic Data

We first consider a simple synthetic experiment environment where $m = 2$ and where the pair (ψ, ξ) is drawn from a mixture of three 4-d multivariate normal distributions. We sample $N = 2000$ observations and use 600 observations to train 400 as validation and 1000 observations for testing. All our results present statistics that are based on 10 simulations, each of which employed a slightly modified mixture model (see Section 2.8.2 for details). The TbS and DTbS algorithms leverage deep neural networks with the corresponding task losses to learn the necessary components $(\mu_\theta(\psi), \Sigma_\theta(\psi))$ of $\mathscr{U}_\theta(\psi)$. All sets are calibrated for a probability coverage of 90% and the risk of decisions is measured using CVaR at risk level $\alpha = 0.9$. We also consider an "oracle" method that leverages the exact knowledge of the underlying distribution as an additional benchmark. The method is based on formulating a scenario tree approximation of the joint distribution of ψ and ξ in order to obtain an investment policy that minimizes the CVaR objective (2.3) directly. More details can be found in Appendix 2.8.3. The average CVaR objective values and marginal coverages of the uncertainty sets can be found in Table 2.1. One can notice that the end-to-end based methods, TbS and DTbS significantly outperform the ETO methods on the CVaR performance. It appears that in order to maintain the required marginal coverage, the ETO approaches learned sets that resulted in overly conservative RO solutions. We also observe that the TbS and DTbS models achieve a CVaR performance that is very close to our estimate of the best achievable performance, i.e. the oracle method's performance.

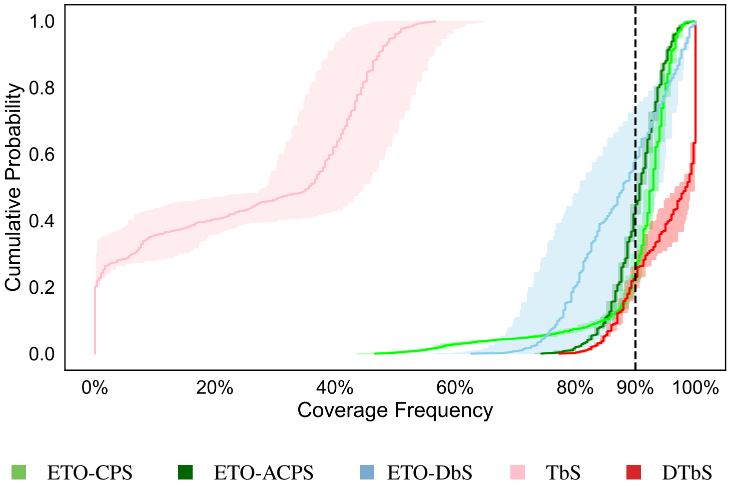


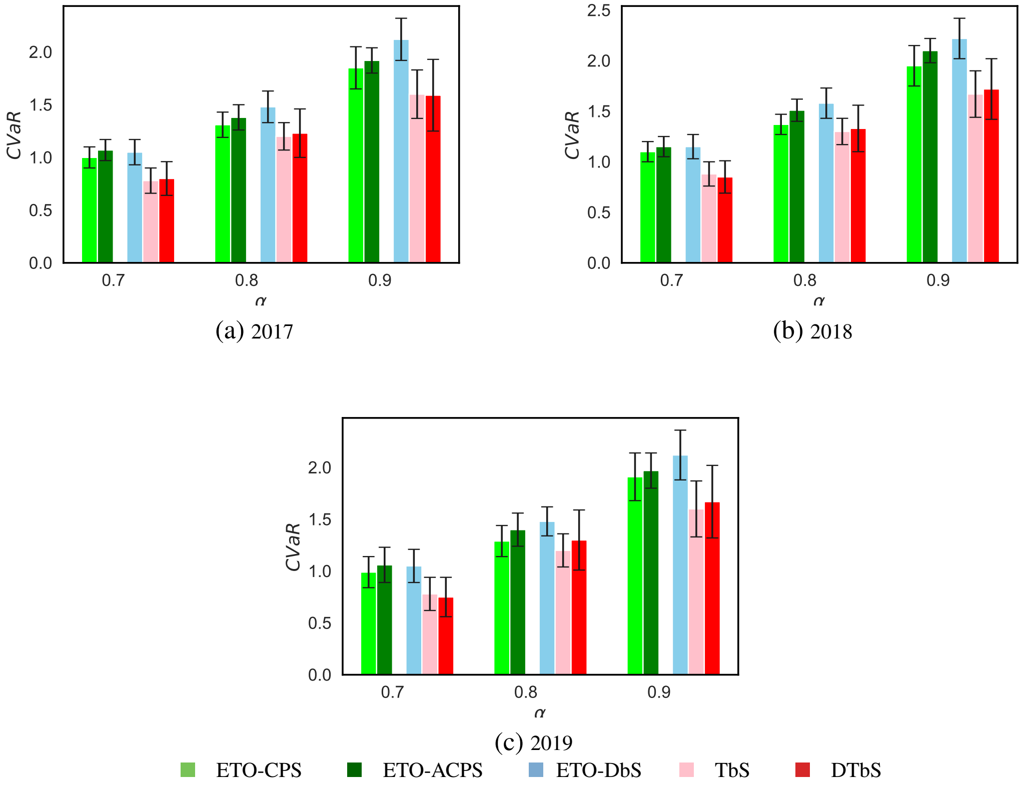
Figure 2.4: Average cumulative distribution of conditional coverage frequency when ψ is sampled uniformly from dataset over 10 simulated environments. Shaded region represent 90% CI

Method	CVaR	Marginal Coverage
ETO-CPS	1.59 ± 0.03	$91 \pm 1.8\%$
ETO-ACPS	1.68 ± 0.04	$91 \pm 1.4\%$
ETO-DbS	1.66 ± 0.06	$85 \pm 7.8\%$
TbS	1.05 ± 0.09	$23 \pm 6.1\%$
DTbS	1.07 ± 0.09	$92 \pm 1.5\%$
Oracle	1.06 ± 0.10	–

Table 2.1: Avg. CVaR and marginal coverage for $\alpha = 1 - \varepsilon = 0.9$ over 10 simulated environments, error represent 90% CI. Note that the oracle method exploits full information about the Gaussian mixture model.

Additionally, all the models except TbS appear to have the marginal coverage 90% which corresponds to the α level they are trained for. By disregarding the aspect of coverage, TbS was able to improve on the CVaR task but performs poorly in terms of coverage. Comparatively, the dual task based approach DTbS was able to improve on the CVaR performance over the ETO approaches while still maintaining the necessary coverage.

As pointed out earlier, conditional coverage is a highly desirable property. Given that a synthetic environment gives us access to exact measurements of conditional coverage, Figure 2.4 presents the cumulative distribution of the observed conditional coverage frequencies when ψ is sampled uniformly from the data set. One can notice from the plot that ETO-DbS, despite being closer to the required marginal coverage, failed to provide accurate conditional coverage. Among the methods that use conformality score to calibrate the radius, ETO-ACPS method which uses localized covariance matrices has better conditional coverage. However, this comes at the price of CVaR performance. The advantages of the dual task-based approach, DTbS, over the single task one are obvious. While DTbS appears to have overshot the coverage compared to ETO-ACPS, which aligns closer to 90%, we argue that this is not an issue as it ends up providing more coverage than needed while generating nearly the best average CVaR value. In Figure 2.3 which overlays the various sets learned on the conditional distribution of ξ, one can notice that the sets adapt to the covariate information ψ to provide the necessary conditional coverage.



(a) 2017

(b) 2018

(c) 2019

Figure 2.5: Avg. CVaR of returns across 10 portfolio trajectory simulations. Error bars report 95% CI.

2.6.3 CRO Using U.S. Stock Data

We follow the experimental design methodology proposed in Chenreddy, Bandi, and Delage (2022). Our experiments utilize historical US stock market data, comprising adjusted daily closing prices for 70 stocks across 8 economical sectors from January 1, 2012, to December 31, 2019, obtained via Yahoo! Finance's API. Each year contains 252 data points, and we calculate percentage gain/loss relative to the previous day to construct our dataset, denoted as ξ. We incorporate the trading volume of individual stocks and other market indices as covariates. We test the robustness of all the model's performance by solving the portfolio optimization problem on randomly selected stock subsets across different periods. Utilizing 15 stocks in each window, we ran the experiment ten times over three moving time frames. We maintain consistent parameters (learning rate lr, number of epochs T, step size K, γ). Further implementation and parameter tuning details can be found in Appendix 2.8.2. Figure 2.5 compares the avg. CVaR of returns and Table 2.2 presents the marginal coverage across different confidence levels for models.

It is evident from the CVaR comparison that the task based methods TbS and DTbS consistently perform better over the ETO models. Among ECRO approaches, we can clearly observe an advantage for DTbS over TbS, which has on par CVaR performance while having out of sample marginal coverage closer to the expected target level. Conformal-based ETO methods have good marginal coverage as they are designed to have the desired coverage. Especially, ETO-ACPS and ETO-CPS, being calibrated using conformal prediction which produces statistically valid prediction regions have near target coverage levels.

2.7 Conclusion

In summary, this chapter introduces a novel framework for conditional robust optimization by combining machine learning and optimization techniques in an end-to-end approach. The study focuses on enhancing the conditional coverage of uncertainty sets and improving CRO performance. Through comparative analysis and simulated experiments, the proposed

Model	Year	Marginal cov. (%)		
		Target $1-\varepsilon$		
		70%	80%	90%
ETO-CPS	2017	68	78	87
ETO-ACPS		68	77	89
ETO-DbS		54	72	85
TbS		22	26	28
DTbS		**72**	**79**	**88**
ETO-CPS	2018	67	79	88
ETO-ACPS		68	78	87
ETO-DbS		59	75	87
TbS		23	24	29
DTbS		**71**	**80**	**93**
ETO-CPS	2019	69	78	88
ETO-ACPS		71	78	89
ETO-DbS		61	76	86
TbS		26	30	32
DTbS		**69**	**78**	**92**

Table 2.2: Marginal coverage (%) evaluated at target coverage levels of 70%, 80%, and 90%. Bold values indicate the best-performing model for each year and target.

methodologies show superior results in robust portfolio optimization. The findings point to the importance of uncertainty quantification and highlight the effectiveness of an end-to-end approach in risk averse decision-making under uncertainty.

Acknowledgments

The authors thank Denis Larocque for helpful discussions. The authors also gratefully acknowledge support from the Institut de Valorisation des Données (IVADO), the Canadian Natural Sciences and Engineering Research Council [RGPIN-2022-05261], and the Canada Research Chair program [CRC-2018-00105].

2.8 Appendix

2.8.1 Algorithms

DTbS algorithm:

```
Algorithm 7 : Dual ECRO Training with Trust Region Solver
Input : Dataset $\mathscr{D}_{\xi,\psi}$; max epochs $T$; max TRO steps $K$; batch size $N$; protection
        level $\alpha$
Initialize warm start buffer $\{\bar{x}_1, \dots, \bar{x}_M\}$ with each $\bar{x}_i \in \mathscr{X}(\psi_i)$
Initialize network parameters $\theta$ and set $t := 1$
while not converged and $t \le T$ do
    Sample a batch of $N$ indices $\mathscr{B} \subset \{1, \dots, M\}$
    for $i \in \mathscr{B}$ do
                                                    // Run TRO for up to $K$ steps
        $(x_i^t, \lambda_i^t, \nu_i^t) \leftarrow \text{TRO}(\bar{x}_i, \mu_\theta(\psi_i), \Sigma_\theta(\psi_i), K)$
        $\bar{x}_i \leftarrow x_i^t$                                 // Update warm start
        $y_i^t \leftarrow \mathbb{I}\{\xi_i \in \mathscr{E}(\mu_\theta(\psi_i), \Sigma_\theta(\psi_i))\}$       // Coverage task label
    $\phi^t \leftarrow$ solve problem (2.8) for $\{(\psi_i, y_i^t)\}_{i \in \mathscr{B}}$
    Compute $\mathscr{L}_{DT}(\theta)$ and $\nabla_\theta \mathscr{L}_{DT}(\theta)$ for $i \sim \mathscr{B}$
    $\theta \leftarrow \theta - \text{step size} \cdot \nabla_\theta \mathscr{L}_{DT}(\theta)$
return $\theta$
```

2.8.2 Supplementary for Experiments

Synthetic Data Generation Process

Our synthetic experiments rely on a set of mixtures of three multivariate normal distributions created in a way that produces a bimodal mixture of a normal distribution with a possibly non-normal one with similar covariance matrix. Specifically, each mixture model is constructed using the same three mean vectors $\mu_a = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T$, $\mu_b = \begin{bmatrix} 0 & 5 & 5 & 0 \end{bmatrix}^T$, and $\mu_c = \mu_b$ while the covariance matrices take the form

$$\Sigma_a = \begin{bmatrix} 1 & 0 & 0.37 & 0 \\ 0 & 1.5 & 0 & 0 \\ 0.37 & 0 & 2 & 0.73 \\ 0 & 0 & 0.73 & 3 \end{bmatrix},$$

$\Sigma_b = \alpha\Sigma_a$ and $\Sigma_c = \frac{\Sigma_a}{\alpha}$ for some $\alpha \in [0,1]$, which controls the non-normality of the second mode. Furthermore, we introduce asymmetry in the mixture model by using the mixing proportion $p_a = \phi$, $p_b = \frac{1-\phi}{\alpha+1}$, and $p_c = \frac{\alpha(1-\phi)}{\alpha+1}$ for some $\phi \in [0,1]$, which controls the dominance of the first mode over the second. Furthermore, p_b and p_c are such that the covariance matrix of the non-normal mixture is equal to the covariance of the normal one, Σ_a.

Synthetic Conditional Data Generation

To generate conditional samples for the synthetic data generated in Section 2.8.2, we first compute the conditional mean $\mu_{\xi|\psi}$ and covariance $\Sigma_{\xi|\psi}$ of ξ given the observed variables ψ for each mixture component. Specifically, for each mean vector μ and covariance matrix Σ associated with the mixture components (denoted as a, b, and c in Section 2.8.2), we calculate the conditional parameters as,

$$\mu_{\xi|\psi} = \mu_\xi + \Sigma_{\xi\psi}\Sigma_{\psi\psi}^{-1}(\psi - \mu_\psi)$$
$$\Sigma_{\xi|\psi} = \Sigma_{\xi\xi} - \Sigma_{\xi\psi}\Sigma_{\psi\psi}^{-1}\Sigma_{\psi\xi}$$

Next, we determine the conditional probability of each mixture given the ψ observation using Bayes theorem as $\mathbb{P}(\text{mixture} = i|\psi) \propto \mathbb{P}(\psi|\text{mixture} = i)\mathbb{P}(\text{mixture} = i)$. Finally, we can use these conditional probabilities to sample new data points from the respective conditional distributions of ξ given ψ.

Parameter Tuning Procedure

In this section, we explore the parameter tuning methodology applied to train the network introduced in Section 2.6.3. Given the time series nature of the data, we employ a rolling window technique for network training. Our architecture depends on a set of hyperparameters, defined as follows: lr for learning rate, T for the maximum number of epochs, K for the maximum TRO steps, B for the batch size, and α for the target level. We partition the data into training and validation periods and examine the optimal combination through grid search. For each combination, we train the network and derive the optimal policy using the training data, then apply it to the unseen validation data. The optimal combination is selected based on the lowest CVaR on the validation dataset, viewing this as a worst-case return minimization problem.

Regarding the DTbS algorithm, which balances between two losses, the CRO objective and the conditional coverage loss, we follow a specific strategy to identify the best-performing model. At each epoch, we save the model and initiate model selection only after achieving the required training coverage. Subsequently, we retain the best models meeting the coverage criteria until convergence conditions are met. Among all saved models meeting the coverage requirement, we choose the one with the best CVaR objective.

Sensitivity analysis:

We conducted a sensitivity analysis of the validation performance as a function of γ, which balances the CVaR loss and the conditional coverage loss. The table below presents the model performances on the validation data for different values of γ. It illustrates how varying γ enables a trade-off between the two loss objectives.

γ	0.01	0.1	0.5	0.9	0.99
avg. $\mathscr{L}_{\text{ECRO}}$	1.30	1.05	1.04	1.06	1.05
avg. $\mathscr{L}_{\text{CC}}$	5.49	6.25	8.15	8.98	8.81

Convergence Comparison

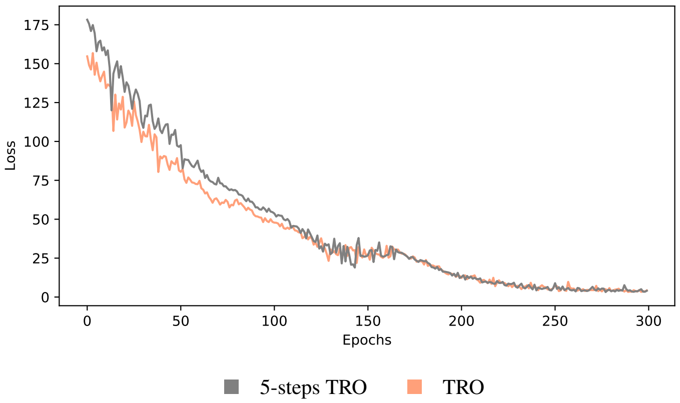
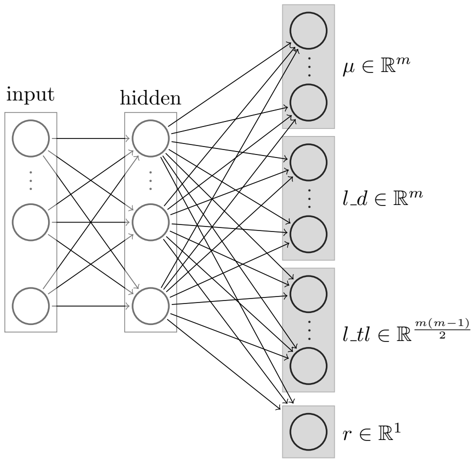


Figure 2.6: Convergence comparison between 5-steps TRO (46 min) and full TRO (129 min).

Architecture



We construct a parametric model for μ and Σ using Cholesky decomposition to ensure positive definiteness of Σ. We employ a shallow neural network architecture with m input units, one hidden layer of size h, and $2m + \frac{m(m-1)}{2} + 1$ units in the output layer. We use tanh for activation functions and softplus for diagonal elements of L to ensure strictly positive values.

2.8.3 Oracle Method for Synthetic Experiments

Given that experiments in Section 2.6.2 are based on a synthetic model, we can evaluate the level of sub-optimality of the portfolio policies proposed by the different models. To do so, we developed an "oracle"-based method that has access to the true underlying joint distribution of ψ and ξ and attempts to identify the "true" optimal value of the CVaR objective, namely

$$\min_{\boldsymbol{x}:\Psi\to\mathscr{X}} \mathrm{CVaR}(-\xi^T x(\psi)).$$

We utilize a scenario tree $\{\psi^i, \{\xi^{ij}\}_{j=1}^M\}_{i=1}^N$ to approximate the joint distribution of (ψ, ξ), where $\psi^i \sim F_\psi$ and $\xi^{ij} \sim F_{\xi|\psi^i}$. Under such scenario tree, the CVaR optimization problem reduces to a linear program:

$$\min_{\{x^i\}_{i=1}^N, \lambda, \{s_{ij}\}_{i=1,j=1}^{N,M}} \quad \lambda + \frac{1}{NM(1-\alpha)} \sum_{i=1}^N \sum_{j=1}^M s_{ij} \tag{2.12a}$$

$$\text{subject to} \quad s_{ij} \geq 0\,,$$

$$\forall i = 1, \ldots, N,\; j = 1, \ldots, M \tag{2.12b}$$

$$s_{ij} \geq -(\xi^{ij})^T x^i - \lambda\,,$$

$$\forall i = 1, \ldots, N,\; j = 1, \ldots, M \tag{2.12c}$$

$$x^i \geq 0\,,\; \forall i = 1, \ldots, N \tag{2.12d}$$

$$\mathbf{1}^T x^i = 1\,,\; \forall i = 1, \ldots, N. \tag{2.12e}$$

To be consistent we the test environment, we consider the $\{\psi_i\}_{i=1}^N$, with $N = 1000$, to take on the values of the test set, while $\{\xi^{ij}\}_{j=1}^M$, for each i with $M = 1000$, are randomly sampled from $F_{\xi|\psi^i}$. This is repeated for the 10 problem instances. The average CVaR optimal value of problem (2.12) is reported in Table 2.1 as the performance of the oracle method.

References

Amos, B., and J. Z. Kolter. 2017. “OptNet: Differentiable optimization as a layer in neural networks.” In *International Conference on Machine Learning,* 70:136–145. PMLR.

Barber, R., E. Candès, A. Ramdas, and R. Tibshirani. 2020. “The limits of distribution-free conditional predictive inference.” *Information and Inference: A Journal of the IMA* 10 (August). https://doi.org/10.1093/imaiai/iaaa017.

Barratt, S., and S. Boyd. 2023. “Covariance prediction via convex optimization.” *Optimization and Engineering* 24 (3): 2045–2078.

Ben-Tal, A., D. Den Hertog, and J.-P. Vial. 2015. “Deriving robust counterparts of nonlinear uncertain inequalities.” *Mathematical programming* 149 (1-2): 265–299.

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

Berthet, Q., M. Blondel, O. Teboul, M. Cuturi, J.-P. Vert, and F. Bach. 2020. “Learning with differentiable pertubed optimizers.” *Advances in neural information processing systems* 33:9508–9519.

Bertsimas, D., and N. Kallus. 2020. “From predictive to prescriptive analytics.” *Management Science* 66 (3): 1025–1044.

Blanquero, R., E. Carrizosa, and N. Gómez-Vargas. 2023. “Contextual Uncertainty Sets in Robust Linear Optimization.”

Blondel, M., Q. Berthet, M. Cuturi, R. Frostig, S. Hoyer, F. Llinares-López, F. Pedregosa, and J.-P. Vert. 2022. “Efficient and modular implicit differentiation.” *Advances in neural information processing systems* 35:5230–5242.

Burton, J. W., M.-K. Stein, and T. B. Jensen. 2020. “A systematic review of algorithm aversion in augmented decision making.” *Journal of Behavioral Decision Making* 33 (2): 220–239.

Butler, A., and R. H. Kwon. 2023. "Efficient differentiable quadratic programming layers: an ADMM approach." *Computational Optimization and Applications* 84 (2): 449–476.

Byrd, R. H., J. C. Gilbert, and J. Nocedal. 2000. "A trust region method based on interior point techniques for nonlinear programming." *Mathematical programming* 89:149–185.

Chenreddy, A. R., N. Bandi, and E. Delage. 2022. "Data-driven conditional robust optimization." *Advances in Neural Information Processing Systems* 35:9525–9537.

Costa, G., and G. N. Iyengar. 2023. "Distributionally robust end-to-end portfolio construction." *Quantitative Finance* 23 (10): 1465–1482.

Donti, P., B. Amos, and J. Z. Kolter. 2017. "Task-based end-to-end model learning in stochastic optimization." *Advances in neural information processing systems* 30.

Duvenaud, D., J. Z. Kolter, and M. Johnson. 2020. "Deep Implicit Layers Tutorial - Neural ODEs, Deep Equilibrium Models, and Beyond." *Neural Information Processing Systems Tutorial.*

Elmachtoub, A. N., and P. Grigas. 2022. "Smart "predict, then optimize"." *Management Science* 68 (1): 9–26.

Gawlikowski, J., C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher, et al. 2021. "A survey of uncertainty in deep neural networks." *arXiv preprint arXiv:2107.03342.*

Gibbs, I., J. J. Cherian, and E. J. Candès. 2023. *Conformal Prediction With Conditional Guarantees.* arXiv: 2305.12616 `[stat.ME]`.

Guo, C., G. Pleiss, Y. Sun, and K. Q. Weinberger. 2017. "On calibration of modern neural networks." In *International conference on machine learning,* 1321–1330. PMLR.

Hannah, L., W. Powell, and D. Blei. 2010. “Nonparametric density estimation for stochastic optimization with an observable state variable.” *Advances in Neural Information Processing Systems* 23.

Johnstone, C., and B. Cox. 2021. “Conformal uncertainty sets for robust optimization.” In *Conformal and Probabilistic Prediction and Applications,* 72–90. PMLR.

Kannan, R., G. Bayraksan, and J. R. Luedtke. 2020. “Data-driven sample average approximation with covariate information.” *Optimization Online. URL: http://www. optimization-online. org/DB HTML/2020/07/7932. html.*

Kontolati, K., D. Loukrezis, D. G. Giovanis, L. Vandanapu, and M. D. Shields. 2022. “A survey of unsupervised learning methods for high-dimensional uncertainty quantification in black-box-type problems.” *Journal of Computational Physics* 464:111313.

Kotary, J., F. Fioretto, P. Van Hentenryck, and B. Wilder. 2021. “End-to-End Constrained Optimization Learning: A Survey.” In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence,* 4475–4482. International Joint Conferences on Artificial Intelligence Organization.

Mandi, J., J. Kotary, S. Berden, M. Mulamba, V. Bucarey, T. Guns, and F. Fioretto. 2024. “Decision-focused learning: Foundations, state of the art, benchmark and future opportunities.” *Journal of Artificial Intelligence Research* 80:1623–1701.

Mandi, J., P. J. Stuckey, T. Guns, et al. 2020. “Smart predict-and-optimize for hard combinatorial optimization problems.” In *Proceedings of the AAAI Conference on Artificial Intelligence,* 34:1603–1610. 02.

Messoudi, S., S. Destercke, and S. Rousseau. 2022. “Ellipsoidal conformal inference for Multi-Target Regression.” In *Conformal and Probabilistic Prediction with Applications,* 294–306. PMLR.

Mišić, V. V., and G. Perakis. 2020. “Data analytics in operations management: A review.” *Manufacturing & Service Operations Management* 22 (1): 158–169.

Ohmori, S. 2021. "A Predictive Prescription Using Minimum Volume k-Nearest Neighbor Enclosing Ellipsoid and Robust Optimization." *Mathematics* 9 (2): 119.

Patel, Y., S. Rayan, and A. Tewari. 2023. "Conformal Contextual Robust Optimization." *arXiv preprint arXiv:2310.10003.*

Qi, M., and M. Shen. 2022. "Integrating Prediction/Estimation and Optimization with Applications in Operations Management," 36–58. October. ISBN: 978-0-9906153-7-8. https://doi.org/10.1287/educ.2022.0249.

Romano, Y., R. F. Barber, C. Sabatti, and E. Candès. 2020. "With malice toward none: Assessing uncertainty via equalized coverage." *Harvard Data Science Review* 2 (2): 4.

Ruszczyński, A., and A. Shapiro. 2021. "Chapter 6: Risk Averse Optimization." In *Lectures on Stochastic Programming: Modeling and Theory, Third Edition,* 223–305. SIAM. https://doi.org/10.1137/1.9781611976595.ch6. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611976595.ch6. https://epubs.siam.org/doi/abs/10.1137/1.9781611976595.ch6.

Sadana, U., A. Chenreddy, E. Delage, A. Forel, E. Frejinger, and T. Vidal. 2025. "A survey of contextual optimization methods for decision-making under uncertainty." *European Journal of Operational Research* 320 (2): 271–289.

Seeger, M., A. Hetzel, Z. Dai, E. Meissner, and N. D. Lawrence. 2017. "Auto-differentiating linear algebra." *arXiv preprint arXiv:1710.08717.*

Sen, S., and Y. Deng. 2018. "Learning enabled optimization: Towards a fusion of statistical learning and stochastic programming." *INFORMS Journal on Optimization (submitted).*

Shafer, G., and V. Vovk. 2008. "A Tutorial on Conformal Prediction." *Journal of Machine Learning Research* 9 (3).

Sun, C., L. Liu, and X. Li. 2023. “Predict-then-Calibrate: A New Perspective of Robust Contextual LP.” In *Advances in Neural Information Processing Systems (NeurIPS).* https://proceedings.neurips.cc/paper_files/paper/2023/file/397271e11322fae8ba7f827c50ca8d9b-Paper-Conference.pdf.

Vovk, V. 2012. “Conditional validity of inductive conformal predictors.” In *Asian conference on machine learning,* 475–490. PMLR.

Wang, I., C. Becker, B. Van Parys, and B. Stellato. 2023. “Learning for Robust Optimization.” *arXiv preprint arXiv:2305.19225.*

Chapter 3

Epistemic Robustness in Offline Reinforcement Learning

Abstract

Offline reinforcement learning aims to learn policies from fixed datasets without further environment interaction. A key challenge in this setting is epistemic uncertainty, which arises from limited or biased data coverage, particularly when the behavior policy systematically avoids certain actions. This can lead to inaccurate value estimates and unreliable generalization. Ensemble-based methods like SAC-N mitigate this by conservatively estimating Q-values using the ensemble minimum, but they require large ensembles and often conflate epistemic with aleatoric uncertainty. To address these limitations, we propose a unified and generalizable framework that replaces discrete ensembles with compact uncertainty sets over Q-values. We further introduce an Epinet based model that directly shapes the uncertainty sets to optimize the cumulative reward under the robust Bellman objective without relying on ensembles. We also introduce a benchmark for evaluating offline RL algorithms under risk-sensitive behavior policies, and demonstrate that our method achieves improved robustness and generalization over ensemble-based baselines across both tabular and continuous state domains.

3.1 Introduction

Offline Reinforcement Learning (RL) aims to learn effective policies from static datasets without further interactions with the environment. A key challenge in this setting is that the uncertainty arises due to insufficient knowledge of the environment, particularly in regions of the state-action space that are poorly represented in the data. This is a prevalent problem in many real world applications where data collection is an inherently costly process. For instance, in personalized healthcare treatment planning or industrial control, collecting large scale interaction data may be impractical or unethical due to cost, safety, or privacy constraints (Ghosh et al. 2022; Levine et al. 2020). This lack of coverage can lead to erroneous value estimates and unreliable generalization, particularly when standard RL algorithms attempt to extrapolate beyond observed data (Y. Yang et al. 2021).

To mitigate this, modern offline RL algorithms such as Soft Actor-Critic with Ensembles (SAC-N) and its variants employ ensembles of Q-networks to quantify uncertainty of the Q-value estimates (An et al. 2021). These methods maintain a collection of N independently initialized but jointly trained critics $\{Q_\theta^{(i)}\}_{i=1}^N$ and construct a conservative Bellman target using the pointwise minimum:

$$y(s,a) := r + \gamma \min_{i\in[N]} Q_\theta^{(i)}(s',a') - \alpha \log \pi_\phi(a'|s'), \tag{3.1}$$

where $(s,a,r,s') \sim \mathscr{D}$, with $\mathscr{D}$ as the dataset, and $a' \sim \pi_\phi(\cdot|s')$ is an action sampled from the policy π_ϕ, parameterized by ϕ, which maps a state s' to a distribution over actions. Here, $\gamma \in (0,1]$ is the discount factor and $\alpha > 0$ controls the entropy of the policy. This formulation treats the minimum over ensemble members as a proxy for a lower confidence bound, promoting conservative estimates in uncertain regions. While empirically effective, ensemble based methods suffer from key limitations. First, reliable uncertainty estimation typically requires large ensemble sizes ($N \geq 10$ in common implementations, and in some cases even hundreds, e.g., $N = 500$ for Hopper-Medium in An et al. (2021)), incurring substantial computational and memory overhead during both training and inference, and limiting scalability in high-dimensional domains (Wen, Tran, and Ba 2020). Second, the pointwise minimum discards inter-action correlations thereby reducing expressivity of the ensembles. Third, ensembles conflate *epistemic* and *aleatoric* uncertainty, making it difficult to disentangle uncertainty due to data scarcity from inherent environmental stochasticity (Amini et al. 2020; Osband et al. 2023). This can hinder robust reasoning about what the agent does not know, and can lead to unsafe or overly conservative policies.

Even in scenarios with abundant data, epistemic uncertainty can persist due to behavioral policy bias, when the data is generated by a policy that systematically favors certain actions (Schweighofer et al. 2022). To illustrate, consider a machine replacement problem (Wiesemann, Kuhn, and Rustem 2013) formulated as a Markov decision process with $|\mathscr{S}| = 10$ states and $|\mathscr{A}| = 2$ actions. At each state $s \in \{1, \ldots, 10\}$, the agent chooses either to continue operation ($a = 1$) or to replace the machine ($a = 2$). Continuing operation increases the chances of reaching a level of severe machine failure. A *risk-averse* behavioral policy may choose to replace early to minimize the chances of reaching the failure state, whereas a *risk-seeking* policy may defer replacement until imminent failure becomes more certain to keep replacement costs to a minimum. Data collection under such fixed policies can result in certain severely underexplored state-action pairs. This sparse coverage leads to erroneous estimation of both the transition dynamics $p(s' \mid s, a)$ and value function $Q(s, a)$. The resulting epistemic uncertainty poses a significant challenge in offline reinforcement learning, where the agent must learn an optimal policy from static data without further environment interaction. Appendix 3.9.1 presents this example in detail, including the optimal policies under varying risk preferences and the resulting state-action visitation distributions under different risk tolerance levels.

To overcome these limitations, we introduce a unified and generalizable alternative that replaces the discrete ensemble $\{Q^{(i)}(s, a)\}_{i=1}^{N}$ with a compact uncertainty set $\mathscr{U}(s) \subset \mathbb{R}^{|\mathscr{A}|}$ defined at each state. This leads to the following set-based Bellman target:

$$y(s,a) := r + \gamma \min_{\mathbf{q} \in \mathscr{U}(s')} \mathbb{E}_{a' \sim \pi_\phi(\cdot|s')} \left[q(a') - \alpha \log \pi_\phi(a'|s') \right], \tag{3.2}$$

where $\mathscr{U}(s')$ compactly models a set of plausible Q-value vectors over actions at state s'. This formulation enables richer representation of uncertainty.

Our key contributions are as follows:

- We propose the Epistemic Robust Soft Actor-Critic (ERSAC) model as an alternative and generalization for the ensemble based SAC-N method. ERSAC exploits uncertainty sets to capture epistemic joint uncertainty about the Q-values of each action, thus enabling richer and more structured epistemic uncertainty modeling
- We integrate epistemic neural network (Epinets) (Osband et al. 2023) in the new ERSAC framework and show how Epinets can be adapted to directly produce uncertainty sets, circumventing the need for resampling at inference time. This latter implementation is shown to scale efficiently to high-dimensional offline RL settings.
- We introduce a benchmark framework for evaluating offline RL algorithms under risk-sensitive behavioral policies, spanning both tabular and continuous state domains. Empirically, our method outperforms ensemble-based baselines across diverse tasks, achieving improved robustness and generalization.

3.2 Related Work

While the **motivation for offline RL** originates primarily from safety, cost, and deployment constraints in domains such as healthcare, robotics, and industrial control, recent work highlights its broader benefits, including improved generalization and sample efficiency when combined with online learning (Ball et al. 2023; Jelley et al. 2024). Offline data can stabilize learning and accelerate convergence through pretraining or regularization (Kumar et al. 2022). However, the absence of environment interaction exacerbates challenges like overestimation and error compounding, especially when using deep value function approximators. These failures are often attributed to epistemic uncertainty in out of distribution state-action pairs, where neural networks are known to make overconfident predictions (Lakshminarayanan, Pritzel, and Blundell 2017; Kendall and Gal 2017). Ensemble-based and Bayesian methods partially mitigate this by explicitly modeling uncertainty, highlighting the need for structured epistemic reasoning in offline settings.

Model-free methods primarily focus on constraining the learned policy or value estimates to remain within the support of the dataset, thereby mitigating extrapolation errors. One class of such methods, known as policy constraint methods, restricts the learned policy to stay close to the behavior policy. This reduces the likelihood of selecting actions not well represented in the data. Approaches like BCQ (Fujimoto, Hoof, and Meger 2018), BEAR (Kumar et al. 2019), and BRAC (Wu, Tucker, and Nachum 2019) explicitly enforce such constraints using divergence penalties or support matching. Another class focuses on value regularization, where conservative value estimates discourage overoptimistic Q-values for out-of-distribution actions. Notably, CQL (Kumar et al. 2020) enforces a soft lower-bound on Q-values, while EDAC (An et al. 2021) and other ensemble-based methods use Q-function diversity to reduce overestimation risk.

Model-based methods instead aim to learn an explicit model of the environment's dynamics, which can be used for policy learning or evaluation via simulated rollouts. Examples include MOPO (Yu et al. 2020), which penalizes uncertainty in model rollouts, and MOReL (Kidambi et al. 2020), which builds a pessimistic MDP based on model confidence. COMBO (Yu et al. 2021) combines model-based rollouts with conservative value estimation to balance optimism and safety.

Other notable directions include trajectory optimization and decision-based methods, such as Decision Transformer (DT) (L. Chen et al. 2021) and Implicit Q-Learning (IQL) (Kostrikov, Nair, and Levine 2021), which cast offline RL as a supervised learning problem over sequences or value distributions. Additionally, imitation-based methods like BAIL (X. Chen et al. 2020) interpolate between behavior cloning and value-based methods using uncertainty-aware selection of demonstration trajectories. We refer the reader to Levine et al. (2020) and Prudencio, Maximo, and Colombini (2023) for comprehensive review of offline RL algorithms.

While **uncertainty quantification** is well studied in supervised learning and Bayesian RL (Ghavamzadeh et al. 2015), its structured application in offline reinforcement learning remains underexplored. Traditional methods often conflate epistemic and aleatoric uncertainty or rely on coarse approximations such as ensemble minima, which can misrepresent uncertainty in regions with limited data. Recent work has begun to address these limitations by introducing methods that model epistemic uncertainty more explicitly. For example, Filos et al. (2022) propose Epistemic Value Estimation (EVE), which provides a task-aware mechanism for quantifying value uncertainty in offline settings. Similarly, Shi and Chi (2022) explore distributionally robust model-based offline RL using uncertainty sets over dynamics to improve robustness to model misspecification. Other approaches such as Panaganti et al. (2022) adopt a risk-sensitive view, incorporating epistemic uncertainty directly into policy optimization to avoid unsafe actions. Ensemble-based methods are a practical way to capture epistemic uncertainty. They have been used in both model-based settings (e.g., MOReL in Kidambi et al. (2020)) and model-free methods (e.g., EDAC in An et al. (2021)) to stabilize learning by regularizing the Bellman backups or penalizing high-variance predictions. However, ensembles can be computationally expensive and coarse. More structured representations of epistemic uncertainty have been proposed using Epistemic Neural Networks (ENNs) (Osband et al. 2023), which offer a flexible way to encode and sample from belief distributions over value functions. Building on these insights, our work introduces a structured, epistemic-robust alternative to ensemble pessimism by defining uncertainty sets over Q-values, allowing richer representations and more targeted conservatism in offline RL.

Additionally, **benchmarking offline RL** remains challenging due to limited dataset diversity. While D4RL (Fu et al. 2020) and RL Unplugged (Gulcehre et al. 2020) have improved standardization, existing benchmarks largely omit risk sensitive evaluation settings. Such behavior policies tend to handle high cost differently depending on whether they are risk averse or risk seeking. This implicit preference skews the data distribution and contributes to epistemic uncertainty, particularly in cases with less data. Despite its significance, there is currently no benchmark that allows systematic control over the risk sensitivity of the behavior policy to study its impact on offline RL performance. As a first step toward addressing this gap, we introduce a framework that enables controlled variation of behavioral risk preferences using dynamic expectiles. This allows us to generate offline datasets with adjustable risk profiles, facilitating principled evaluation of offline RL algorithms under different uncertainty conditions. Our proposed framework is aligned with recent efforts like the Minari platform proposed by Younis et al. (2024), but uniquely focuses on how risk sensitivity shapes epistemic uncertainty in offline datasets.

3.3 Preliminaries

We consider a Markov Decision Process (MDP) characterized by a possibly continuous state space $\mathscr{S}$, a discrete action space $\mathscr{A}$, a state-transition distribution $p(s_{t+1}|s_t, a_t)$, a reward function $r(s_t, a_t)$, and a discount factor $\gamma \in (0,1)$. The reinforcement learning objective is to identify an optimal (possibly random) policy $\pi^*(\cdot|s)$, with $\pi^*(a|s)$ defining the likelihood of doing action a when in state s, that maximizes the expected discounted cumulative reward:

$$\mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right].$$

Below, we summarize the Soft Actor-Critic (SAC) Algorithm and one of its adaptations for offline RL that performs conservative updates using an ensemble of Q-functions.

3.3.1 Soft Actor-Critic Algorithm (SAC)

We adopt Soft Actor-Critic (SAC), originally developed for continuous action spaces, and adapt it to discrete actions (Christodoulou 2019). By introducing entropy regularization, SAC strikes a balance between exploration and exploitation. Formally, for discrete actions, the SAC objective is:

$$J(\pi) := \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t \left(r(s_t, a_t) + \alpha \mathscr{H}(\pi(\cdot|s_t)) \right) \right], \tag{3.3}$$

where the entropy $\mathscr{H}(\pi(\cdot|s_t))$ is defined as

$$\mathscr{H}(\pi(\cdot|s_t)) := -\sum_{a \in \mathscr{A}} \pi(a|s_t) \log \pi(a|s_t),$$

and α is a temperature hyperparameter that controls the influence of the entropy term as a regularizer promoting policy stochasticity. We further define a Q-function $Q(s,a)$, estimating the entropy regularized expected cumulative reward from a state-action pair (s,a) under policy π:

$$Q(s,a) := \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty}\gamma^{t}\Big(r(s_t,a_t)+\alpha\mathscr{H}\big(\pi(\cdot\mid s_t)\big)\Big)\;\Big|\; s_0=s,\ a_0=a.\right] \tag{3.4}$$

We typically represent the Q-function using a parametric model $Q_\theta(s,a)$, e.g. a neural network, to effectively handle continuous and high-dimensional state spaces. The policy $\pi_\phi(a|s)$, parameterized by ϕ, is defined as a categorical probability distribution over discrete actions conditioned on continuous states, facilitating straightforward computation of entropy terms.

The parameters of the policy and Q-functions are updated iteratively using off-policy experiences drawn from a replay buffer $\mathscr{D}$. Specifically, the Q-function parameters θ are updated to minimize the temporal difference (TD) error:

$$\theta \leftarrow \theta - \eta_Q \nabla_\theta \mathbb{E}_{(s,a,r,s')\sim\mathscr{D}}\left[\Big(Q_\theta(s,a) - \big(r+\gamma\mathbb{E}_{a'\sim\pi(\cdot|s')}[Q_{\theta'}(s',a') - \alpha\log\pi_\phi(a'\mid s')]\big)\Big)^2\right]$$

where η_Q is the Q-function learning rate, and θ' denotes parameters of a target Q-network periodically synchronized with θ to enhance training stability. The policy parameters ϕ are updated to maximize the entropy-regularized expected Q-values:

$$\phi \leftarrow \phi + \eta_\pi \nabla_\phi \mathbb{E}_{s\sim\mathscr{D},a\sim\pi_\phi(\cdot|s)}[Q_\theta(s,a) - \alpha\log\pi_\phi(a|s)],$$

where η_π is the policy learning rate.

3.3.2 SAC with an Ensemble of Q-functions (SAC-N)

While Soft Actor-Critic provides a stable framework for policy learning, its direct application to offline reinforcement learning is challenging as the agent must learn solely from a fixed dataset without further interaction with the environment. As a result, standard SAC algorithms are prone to overestimation bias, which arises when the learned Q function extrapolates inaccurately to out-of-distribution state-action pairs. This is particularly problematic in the policy improvement step in SAC, which explicitly encourages the selection of actions that maximize Q-values, amplifying the impact of overestimated values and potentially steering the policy toward suboptimal or unsafe regions of the state-action space. To address this, An et al. (2021) proposed the SAC-N model that maintains an ensemble of N Q-functions, $\{Q_{\theta_i}\}_{i=1}^{N}$, to capture epistemic uncertainty and mitigate overestimation bias. Each Q-function Q_{θ_i} represents an independent estimate of the expected return for a given state-action pair. To stabilize training and further reduce bias, SAC-N uses a target Q-network ensemble $\{Q_{\theta'_i}\}_{i=1}^{N}$, where each θ'_i is updated via Polyak averaging of the corresponding online parameters θ_i. The target used for the Q-function update is based on the clipped double Q-learning objective (Fujimoto, Hoof, and Meger 2018), with SAC-N extending it by considering the minimum over the Q-functions ensemble as,

$$y(r, s', a') := \ r + \gamma\left(\min_{i=1,\ldots,N} Q_{\theta'_i}(s', a') - \alpha \log \pi_\phi(a' \mid s')\right), \tag{3.5}$$

The use of the minimum over the ensemble acts as a conservative estimate of the expected return, reducing the likelihood of propagating overestimated values from out-of-distribution state-action pairs that are common in offline datasets. Each Q-function Q_{θ_i} is updated by minimizing the mean squared Bellman error between its predicted value and the target $y(r, s', a')$:

$$\mathscr{L}_Q(\theta_i) := \mathbb{E}_{(s,a,r,s')\sim\mathscr{D},\, a'\sim\pi_\phi(\cdot|s')}\left[\left(Q_{\theta_i}(s,a) - y(r, s', a')\right)^2\right], \tag{3.6}$$

where $\mathscr{D}$ denotes the static replay buffer of environment interactions, which, unlike in online RL, is fixed and is collected a priori without further interactions. The policy π_ϕ is then optimized to maximize the conservative estimate of the expected return, given by the minimum Q-value across the ensemble, while incorporating the entropy regularization term:

$$\mathscr{J}\pi(\phi) := \mathbb{E}_{s\sim\mathscr{D}, a\sim\pi_\phi(\cdot|s)}\left[\min_{i=1,\ldots,N} Q_{\theta_i}(s,a) - \alpha \log \pi_\phi(a|s)\right]. \tag{3.7}$$

This objective encourages the policy to achieve a trade-off between maximizing a conservative estimate of expected returns and maintaining high entropy. Higher entropy promotes stochasticity in action selection, allowing the policy to occasionally choose actions that are less frequent in the offline dataset. This behavior is particularly beneficial in the early stages of training, where increased randomness helps avoid overfitting to spurious correlations in the data. To gradually shift the focus to maximize the rewards during the training process, we follow Haarnoja et al. (2018) and learn the entropy coefficient α by minimizing a dual objective that encourages the policy entropy to match a target value. This approach allows the agent to maintain high entropy when uncertainty is high and gradually shift focus to reward maximization as learning progresses.

Although SAC-N mitigates overestimation by maintaining an ensemble of Q-functions, it often requires a large ensemble size for stable performance. To address this, An et al. (2021) introduced the **Ensemble-Diversified Actor-Critic (EDAC)**, which adds a diversification term to encourage diversity among the Q-function ensemble members. In continuous action setting, they quantify similarity using an ensemble similarity (ES) metric defined as:

$$\frac{\langle \nabla_a Q_{\theta_i}(s,a), \nabla_a Q_{\theta_j}(s,a) \rangle}{\|\nabla_a Q_{\theta_i}(s,a)\| \|\nabla_a Q_{\theta_j}(s,a)\|},$$

which measures the cosine similarity between the gradients of different Q-functions with respect to the action vector. In the discrete action setting, where $\nabla_a Q(s,a)$ is ill defined, we adapt the ES metric by instead computing the mean squared deviation between the Q-values across all actions. Specifically, we define $g_\theta(s,a) := \left(Q_\theta(s,a') - Q_\theta(s,a)\right)_{a' \in \mathscr{A}}$, and compute the cosine similarity between $g_{\theta_i}(s,a)$ and $g_{\theta_j}(s,a)$:

$$\mathrm{ES}_{\theta_i,\theta_j}(s,a) := \frac{\sum_{a' \in \mathscr{A}} \left(Q_{\theta_i}(s,a') - Q_{\theta_i}(s,a)\right)\left(Q_{\theta_j}(s,a') - Q_{\theta_j}(s,a)\right)}{\sqrt{\sum_{a' \in \mathscr{A}} \left(Q_{\theta_i}(s,a') - Q_{\theta_i}(s,a)\right)^2} \sqrt{\sum_{a' \in \mathscr{A}} \left(Q_{\theta_j}(s,a') - Q_{\theta_j}(s,a)\right)^2}}.$$

The diversification loss is then given by:

$$\mathscr{L}_{\mathrm{ES}}(\theta) := \mathbb{E}_{(s,a)\sim\mathscr{D}} \left[\sum_{i=1}^{N} \sum_{j=i+1}^{N} \mathrm{ES}_{\theta_i,\theta_j}(s,a) \right].$$

where θ is short for $\{\theta_i\}_{i=1}^N$. The overall loss for each Q-function incorporates this diversification term:

$$\bar{\mathscr{L}}_Q(\theta) := (1/N) \sum_{i=1}^{N} \mathscr{L}_Q(\theta_i) + \eta \mathscr{L}_{\mathrm{ES}}(\theta), \tag{3.8}$$

where η is a hyperparameter controlling the strength of the diversity regularization. Encouraging diversity among the Q-functions was shown empirically to improve uncertainty estimation and leads to more reliable policy learning.

3.4 Epistemic Robustness with SAC

Ensemble based approaches, as discussed in Section 3.3.2, model the epistemic uncertainty in value estimation using a finite collection of Q-functions, $\{Q_{\theta_i}\}_{i=1}^{N}$. Each Q_{θ_i} encodes a different hypothesis about the long term expected return, and conservative estimates are obtained by taking the minimum over the ensemble $\min_{i=1,\ldots,N} Q_{\theta_i}(s',a')$. To formalize the uncertainty captured by such an ensemble, we model the long term actions values at a given state s as a distribution $F^q_\theta(s) \in \mathscr{M}(\mathbb{R}^{|\mathscr{A}|})$. Here, $F^q_\theta(s)$ defines a probability measure over Q-value vectors $q \in \mathbb{R}^{|\mathscr{A}|}$, induced by the variability among the Q-functions, and parameterized through θ. Each sample $\tilde{q} \sim F^q_\theta(s)$ is a vector in $\mathbb{R}^{|\mathscr{A}|}$ representing the epistemic uncertainty about the action-wise values $Q(s,\cdot)$. For example, in the case of SAC-N, this distribution takes the form of a scenario-based distribution:

$$F^q_\theta(s) := \frac{1}{N}\sum_{i=1}^{N} \delta_{Q_{\theta_i}(s,\cdot)}, \tag{3.9}$$

where δ_x is the Dirac measure centered at $x \in \mathbb{R}^{|\mathscr{A}|}$. Thus obtaining that $F^q_\theta(s)$ is the distribution of $\tilde{q} := Q_{\theta_{\tilde{i}}}(s,\cdot)$ with $\tilde{i} \sim U(N)$, i.e. the uniform distribution over $1,\ldots,N$.

Given a Q-value distribution $F^q_\theta : \mathscr{S} \to \mathscr{M}(\mathbb{R}^{|\mathscr{A}|})$, which maps each state $s \in \mathscr{S}$ to a probability measure over Q-value vectors, one can define an uncertainty set operator:

$$\mathscr{U} : \mathscr{M}(\mathbb{R}^{|\mathscr{A}|}) \to \mathscr{C}(\mathbb{R}^{|\mathscr{A}|}),$$

that maps a Q-value distribution to a compact set of plausible Q-value vectors. The composition $\mathscr{U} \circ F^q_\theta : \mathscr{S} \to \mathscr{C}(\mathbb{R}^{|\mathscr{A}|})$ defines an epistemic uncertainty set $\mathscr{U}(F^q_\theta(s))$ in each state s, which can be used to construct robust evaluation and optimization of policies. For notational simplicity, we will use $\mathscr{U}_\theta(s)$ as shorthand for $\mathscr{U}(F^q_\theta(s))$ when the dependencies on F^q_θ are clear from context.

This set captures uncertainty around the predicted Q-values and can take various forms depending on the modeling assumptions. For instance, if $F^q_\theta(s)$ is Gaussian, the corresponding uncertainty set $\mathscr{U}(F^q_\theta(s))$ may be defined as an ellipsoidal confidence region centered at the mean and shaped by the covariance matrix. In the ensemble setting, where $F^q_\theta(s)$ is a discrete distribution over Q-functions, $\mathscr{U}(F^q_\theta(s))$ can be constructed again as a confidence region or even as the convex hull enclosing the ensemble realizations, i.e. the distribution's support.

In the next section, we introduce our proposed framework, Epistemic Robust Soft Actor-Critic (ERSAC), a framework that generalizes SAC-N by incorporating the uncertainty sets constructed from the distribution over Q-values. We begin with a version of ERSAC that operates with an ensemble of N Q-functions and establish the connections between ERSAC and SAC-N methods. We then formalize the ERSAC algorithm, outlining its key components, including the set-based Bellman backup and robust policy update.

3.4.1 The Epistemic Robust SAC (ERSAC) Model

Similarly as for SAC-N, our epistemic robust SAC algorithm trains the Q-function by minimizing the expected squared Bellman error between the sampled realization and a conservatively estimated target value measured using the distribution F^q_θ over Q-functions. Specifically, for each next state $s' \in \mathscr{S}$, the robust target value in equation (3.5) is first modified to,

$$y(r,s') := r + \gamma\Bigg(\min_{q \in \mathscr{U}(F^q_{\theta'}(s'))} \mathbb{E}_{a' \sim \pi_\phi(\cdot|s')}\left[q(s',a') - \alpha \log \pi_\phi(a' \mid s')\right]\Bigg), \tag{3.10}$$

where the minimum operator provides a robust estimate of the regularized expected total discounted return and can be calculated using the support function associated to $\mathscr{U}(F^q_{\theta'}(s'))$:

$$\min_{q \in \mathscr{U}(F^q_{\theta'}(s'))} \mathbb{E}_{a' \sim \pi_\phi(\cdot|s')}[q(s',a')] = -\delta^*(-\pi_\phi(\cdot \mid s')|\mathscr{U}_{\theta'}(s'))$$

with $\delta^*(v|\mathscr{U}) := \sup_{q \in \mathscr{U}} \langle v, q \rangle$. We refer the reader to Ben-Tal, Den Hertog, and Vial (2015) for closed form expressions of $\delta^*(v|\mathscr{U})$ for a list of popular forms of uncertainty sets. The loss function in (3.6) is then redefined as:

$$\mathscr{L}_Q^R(\theta) := \mathbb{E}_{(s,a,r,s')\sim\mathscr{D},\ \tilde{q}\sim F_\theta^q(s)} \left[\left(\tilde{q}(a) - y(r,s') \right)^2 \right]. \tag{3.11}$$

It is important to note that without additional regularization, the Objective in (3.11) may admit a degenerate solution $F_{\theta^*}^q(s) = \delta_{\bar{q}(s,\cdot)}$, where $\bar{q}(s,a) := \mathbb{E}_{(s,a,r,s')\sim\mathscr{D}}[y(r,s')]$, which collapses the distribution to a deterministic point estimate. In practice, this requires regularization strategies such as early stopping, entropy constraints on F_θ^q, or prior-based regularization to avoid mode collapse.

Similar to the Q-value target, the policy loss in the epistemic robust setting replaces the ensemble minimum with a worst-case expectation over the uncertainty set. The robust policy loss in equation (3.7) becomes:

$$\begin{aligned} \mathscr{J}_\pi^R(\phi) :=\ & \mathbb{E}_{s\sim\mathscr{D}} \left[\min_{q\in\mathscr{U}_\theta(s)} \mathbb{E}_{a\sim\pi_\phi(\cdot|s)} \left[q(a) - \alpha \log \pi_\phi(a \mid s) \right] \right] \\ =\ & \mathbb{E}_{s\sim\mathscr{D},a\sim\pi_\phi(\cdot|s)} \left[\min_{q\in\mathscr{U}_\theta(s)} \langle \pi_\phi(\cdot \mid s), q \rangle - \alpha \log \pi_\phi(a \mid s) \right]. \end{aligned} \tag{3.12}$$

Importantly, when using an ensemble based representation, the ERSAC formulation encompasses SAC-N as a special case under a particular choice of uncertainty set. We formalize this connection in the following proposition and defer the proof to Appendix 3.9.2.

Proposition 3.4.1. *Let $F_\theta^q(s)$ be defined as in equation (3.9), and let the uncertainty set operator be defined as*

$$\mathscr{U}_{box}(F_\theta^q(s)) := \times_{a\in\mathscr{A}} \left[essinf_{\tilde{q}\sim F_\theta^q(s)}[\tilde{q}(a)],\ esssup_{\tilde{q}\sim F_\theta^q(s)}[\tilde{q}(a)] \right], \tag{3.13}$$

i.e., a coordinate-wise box containing the support of $F_\theta^q(s)$. Then, the robust Q-loss and policy loss reduce to the SAC-N losses:

$$\mathscr{L}_Q^R(\theta) = \frac{1}{N} \sum_{i=1}^{N} \mathscr{L}_Q(\theta_i) + C \quad \text{and} \quad \mathscr{J}_\pi^R = \mathscr{J}_\pi,$$

for some constant $C \in \mathbb{R}$ that is independent of θ.

This result demonstrates that ERSAC generalizes SAC-N under a unified uncertainty set framework. In the next section, for any compact set representation $\mathscr{U}_\theta(s)$, we outline the detailed training algorithm.

3.4.2 The ERSAC Training Algorithm

In the earlier case, we modeled $F^q_\theta(s)$ such that each sample $\tilde{q} \sim F^q_\theta(s)$ is a vector in $\mathbb{R}^{|\mathscr{A}|}$ that represents the action-wise values $Q(s,\cdot)$. To place this representation in a more general framework, we consider a reparametrized formulation as introduced in the Assumption 3.4.1. This formulation includes the ensemble case as a special instance where the noise variable z indexes a finite set of Q-functions. More generally, this formulation admits expressive stochastic representations with both discrete and continuous support.

Assumption 3.4.1. *F^q_θ is associated to a sampling operator $\mathfrak{q}_\theta(s,a,z)$ and a distribution $F_z \in \mathscr{M}(\mathbb{R}^{d_z})$, such that $\mathfrak{q}_\theta(s,\cdot,\tilde{z})$ is distributed according to $F^q_\theta(s)$ when $\tilde{z} \sim F_z$.*

Given a noise sample $\tilde{z} \sim F_z$, a corresponding Q-vector sample $\tilde{q} \sim F^q_\theta(s)$ is obtained by evaluating the sampling operator over all actions:

$$\tilde{q}(a) := \mathfrak{q}_\theta(s,a,\tilde{z}), \quad \text{for all } a \in \mathscr{A}.$$

Formally, $\tilde{q} = \mathfrak{q}_\theta(s,\cdot,\tilde{z}) \in \mathbb{R}^{|\mathscr{A}|}$ is a realization from the epistemic Q-distribution $F^q_\theta(s)$, induced via the sampling operator with epistemic variability governed by the latent variable $\tilde{z} \sim F_z$. This reparameterized formulation subsumes the ensemble-based model described in equation 3.9 as a special case, where the latent variable $\tilde{z} \in \{1,\dots,N\}$ indexes a finite set of Q-functions, and the sampling operator reduces to $q_\theta(s,a,\hat{z}) = Q_{\theta_{\tilde{z}}}(s,a)$.

In order to minimize $\mathscr{L}^R_Q$, when Assumption 3.4.1 is satisfied, one can use a popular reparametrization trick to derive a gradient for the critic parameters θ as:

$$\begin{aligned}\nabla_\theta \mathscr{L}^R_Q(\theta) &= \nabla_\theta \mathbb{E}_{(s,a,r,s')\sim\mathscr{D},\tilde{z}\sim F_z}\left[(\mathfrak{q}_\theta(s,a,\tilde{z}) - y(r,s'))^2\right] \\ &= \mathbb{E}_{(s,a,r,s')\sim\mathscr{D},\tilde{z}\sim F_z}\left[2(\mathfrak{q}_\theta(s,a,\tilde{z}) - y(r,s')) \cdot \nabla_\theta \mathfrak{q}_\theta(s,a,\tilde{z})\right]. \qquad (3.14)\end{aligned}$$

This gives rise to the stochastic update:

$$\theta \leftarrow \theta - \eta_Q \cdot 2\left(\mathfrak{q}_\theta(s,a,\tilde{z}) - y(r,s')\right) \cdot \nabla_\theta \mathfrak{q}_\theta(s,a,\tilde{z}).$$

The question of optimizing $\mathscr{J}^R_\pi$ is a bit more complex. We start by letting $q^*(s,\cdot;\phi)$ be any statewise adversarial Q-value vector for policy π_ϕ:

$$q^*(s,\cdot;\phi) \in \arg\min_{q\in\mathscr{U}_\theta(s)} \langle \pi_\phi(\cdot \mid s), q\rangle, \ \forall s \in \mathscr{S}, \qquad (3.15)$$

which is well-defined due to compactness of $\mathscr{U}_\theta(s)$. Then, noting that the function

$$\begin{aligned} f(\pi) :&= \mathbb{E}_{s\sim\mathscr{D}, a\sim\pi(\cdot|s)}\left[\min_{q\in\mathscr{U}_\theta(s)} \langle \pi(\cdot \mid s), q\rangle - \alpha \log \pi(a \mid s)\right] \\ &= \mathbb{E}_{s\sim\mathscr{D}}\left[\min_{q\in\mathscr{U}_\theta(s)} \langle \pi(\cdot \mid s), q\rangle - \alpha \mathbb{E}_{a\sim\pi(\cdot|s)}\left[\log \pi(a \mid s)\right]\right] \end{aligned}$$

is concave with respect to π, one can invoke the envelope theorem to identify one of its supergradients as

$$\nabla_\pi \mathbb{E}_{s\sim\mathscr{D}}\left[\langle \pi(\cdot \mid s), q^*(s,\cdot;\phi)\rangle - \alpha \mathbb{E}_{a\sim\pi(\cdot|s)} \log \pi(a \mid s)\right] \in \nabla_\pi f(\pi)$$

We therefore obtain, fixing $\bar{\phi}$ to ϕ that:

$$\begin{aligned} \nabla_\phi \mathscr{J}_\pi^R(\phi) :&= \mathbb{E}_{s\sim\mathscr{D}}\big[\nabla_\phi \langle \pi_\phi(\cdot \mid s), q^*(s,\cdot;\bar{\phi})\rangle - \alpha \nabla_\phi \mathbb{E}_{a\sim\pi_\phi(\cdot|s)}\left[\log \pi_\phi(a \mid s)\right]\big] \\ &= \mathbb{E}_{s\sim\mathscr{D}}\big[\sum_{a\in\mathscr{A}} q^*(s,a;\phi)\nabla_\phi \pi_\phi(a \mid s) - \alpha \nabla_\phi \langle \pi_\phi(\cdot \mid s), \log \pi_\phi(\cdot \mid s)\rangle\big]. \quad (3.16) \end{aligned}$$

This produces a standard entropy-regularized policy gradient, but is evaluated with respect to the worst-case value vector $q^*(s,\cdot;\phi)$ in the uncertainty set, providing robustness to epistemic uncertainty. We summarize the training procedure for Robust SAC-N in Algorithm 8.

3.5 Sample Based Construction of $\mathscr{U}_\theta(s)$ from $\mathsf{q}_\theta(s,a,\tilde{z})$

In Section 3.4, we introduced a robust SAC-N framework in which Bellman backups are computed using uncertainty sets $\mathscr{U}_\theta(s)$ derived from distributions $F_\theta^q(s)$ over Q-values. While this formulation assumes access to the full distribution, often one can only approximate $F_\theta^q(s)$ using Monte-Carlo samples, which form an empricial distribution $\widehat{F}_\theta^q(s)$. Having access to $\widehat{F}_\theta^q(s)$, one can approximate $\mathscr{U}(F_\theta^q(s))$ with $\mathscr{U}(\widehat{F}_\theta^q(s))$.

In practice, constructing the uncertainty set $\mathscr{U}_\theta(s)$ from the empirical distribution $\widehat{F}_\theta^q(s)$ requires choosing a specific set operator that defines the shape and inductive bias of the epistemic uncertainty representation. Different choices of $\mathscr{U}(\widehat{F}_\theta^q(s))$ lead to varying trade-offs between computational tractability, policy sensitivity, and expressiveness. In

Algorithm 8 : Epistemic Robust SAC Training

```
Input : Initial policy parameters $\phi$, Q parameters $\theta$, target Q parameters $\theta'$,
        offline data replay buffer $\mathscr{D}$, learning rates $\eta_Q, \eta_\pi$, target update rate $\tau$
for each epoch do
    Sample minibatch $\mathscr{B} := \{(s,a,r,s')\}$ from $\mathscr{D}$
    Compute target:
        $y(r,s') \leftarrow r + \gamma \left( \min_{q \in \mathscr{U}_{\theta'}(s')} \langle \pi_\phi(\cdot|s'), q \rangle - \alpha \mathbb{E}_{a' \sim \pi_\phi}[\log \pi_\phi(a'|s')] \right)$
    Critic update:
        $\theta \leftarrow \theta - \eta_Q \cdot 2 \frac{1}{|\mathscr{B}|} \sum_{(s,a,r,s') \in \mathscr{B}} \mathbb{E}_{\tilde{z} \sim F_z} \left[ (\mathfrak{q}_\theta(s,a,\tilde{z}) - y(r,s')) \cdot \nabla_\theta \mathfrak{q}_\theta(s,a,\tilde{z}) \right]$
    Compute worst-case $q^*$ vectors:
        $q^*(s,\cdot;\phi) \leftarrow \arg \min_{q \in \mathscr{U}_\theta(s)} \langle \pi_\phi(\cdot \mid s), q \rangle$
    Actor update:
        $\phi \leftarrow \phi + \eta_\pi \cdot \frac{1}{|\mathscr{B}|} \sum_{s \in \mathscr{B}} \left( \sum_{a \in \mathscr{A}} q^*(s,a;\phi) \nabla_\phi \pi_\phi(a \mid s) - \alpha \nabla_\phi \mathbb{E}_{a \sim \pi_\phi(a|s)}[\log \pi_\phi(\cdot \mid s)] \right)$
    Update target network: $\theta' \leftarrow \tau\theta + (1-\tau)\theta'$
```

the remainder of this section, we present three popular sets from the literature of robust optimization: box set, convex hull set and ellipsoidal set. Each of these constructions induces a distinct worst-case Q-vector $q^*(s,\cdot;\phi)$, shaping the Bellman backup in different ways. We formalize each construction below and analyze their implications for robust policy evaluation and learning.

3.5.1 Box Set

Let $\{\tilde{z}_i\}_{i=1}^N$ be N values sampled from F_z. The simplest construction is the box set introduced in equation (3.13), which defines $\mathscr{U}_\theta(s)$ as the Cartesian product of the intervals covering $\tilde{q}(a)$ for each action. In a sample-based setting, this reduces to :

$$\mathscr{U}_{\text{box}}(\widehat{F}^q_\theta(s)) := \times_{a\in\mathscr{A}}\left[\min_{i=1,\ldots,N}\mathfrak{q}_\theta(s,a,\tilde{z}_i),\ \max_{i=1,\ldots,N}\mathfrak{q}_\theta(s,a,\tilde{z}_i)\right]. \tag{3.17}$$

This construction treats Q-values for each action as independent and corresponds to the uncertainty model used in SAC-N. However, it produces a worst-case Q-vector that is insensitive to changes in the policy due to its over-conservative, i.e. $q^*(s,a;\phi) = \min_{i\in[N]}\mathfrak{q}(s,a,\tilde{z}_i)$ (see proof of Lemma 3.4.1) independently of ϕ.

3.5.2 Convex Hull Set

A more expressive alternative is the uncertainty set operator that produces the convex hull of the support of $F^q_\theta(s)$. In a sample-based setting, this reduces to:

$$\mathscr{U}_{\text{hull}}(\widehat{F}^q_\theta(s)) := \left\{\sum_{i=1}^N \lambda_i\mathfrak{q}_\theta(s,\cdot,\tilde{z}_i)\,\middle|\,\exists\lambda\in\mathbb{R}^N,\ \lambda_i\geq 0\ \forall i=1,\ldots,N,\ \sum_{i=1}^N\lambda_i = 1\right\}. \tag{3.18}$$

This set captures all convex combinations of the sampled Q-values and preserves dependencies between actions. The worst-case Q-vector takes the form: $q^*(s,a;\phi) = \mathfrak{q}_\theta(s,a,z^*(s,\phi))$ with $z^*(s,\phi)\in\operatorname{argmin}_i\mathbb{E}_{a\sim\pi_\phi(\cdot|s)}[\mathfrak{q}_\theta(s,a,\tilde{z}_i)]$. This is due to:

$$\begin{aligned}
\min_{q\in\mathscr{U}_{\text{hull}}(\widehat{F}^q_\theta(s))}\mathbb{E}_{a\sim\pi_\phi(\cdot|s)}[q(a)] &= \min_{\lambda\geq 0:\sum_{i=1}^N\lambda_i=1}\mathbb{E}_{a\sim\pi_\phi(\cdot|s)}[\sum_{i=1}^N\lambda_i\mathfrak{q}_\theta(s,a,\tilde{z}_i)]\\
&= \min_{\lambda\geq 0:\sum_{i=1}^N\lambda_i=1}\sum_{i=1}^N\lambda_i\mathbb{E}_{a\sim\pi_\phi(\cdot|s)}[\mathfrak{q}_\theta(s,a,\tilde{z}_i)]\\
&\geq \min_{i\in[N]}\mathbb{E}_{a\sim\pi_\phi(\cdot|s)}[\mathfrak{q}_\theta(s,a,\tilde{z}_i)] = \mathbb{E}_{a\sim\pi_\phi(\cdot|s)}[\mathfrak{q}_\theta(s,a,z^*(s,\phi))].
\end{aligned}$$

3.5.3 Ellipsoidal Set

In this work, we will mainly consider an ellipsoidal set operator that aim to cover a certain proportion υ of the total mass of $F^q_\theta(s)$. In a sample-based setting, this can be done by estimating the empirical mean and covariance of the sampled Q-vectors:

$$\hat{\mu}(s) := \frac{1}{N}\sum_{i=1}^N\mathfrak{q}_\theta(s,\cdot,\tilde{z}_i),\quad \widehat{\Sigma}(s) := \frac{1}{N}\sum_{i=1}^N(\mathfrak{q}_\theta(s,\cdot,\tilde{z}_i)-\mu(s))(\mathfrak{q}_\theta(s,\cdot,\tilde{z}_i)-\mu(s))^\top.$$

and estimating the radius as

$$\widehat{\Upsilon}(s) := \inf\{\Upsilon | \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{(\mathfrak{q}_\theta(s,\cdot,\tilde{z}_i) - \hat{\mu}(s))^\top \widehat{\Sigma}(s)^{-1}(\mathfrak{q}_\theta(s,\cdot,\tilde{z}_i) - \hat{\mu}(s)) \leq \Upsilon^2\} \geq \upsilon\}$$

The corresponding uncertainty set is defined as:

$$\mathscr{U}_{\mathrm{ell}}(\widehat{F}^q_\theta(s)) := \left\{ q \in \mathbb{R}^{|\mathscr{A}|} \,\middle|\, (q - \hat{\mu}(s))^\top \widehat{\Sigma}(s)^{-1}(q - \hat{\mu}(s)) \leq \widehat{\Upsilon}(s)^2 \right\} \tag{3.19}$$

This set encodes second-order structure and supports efficient optimization. Indeed, when $\widehat{\Sigma}(s)$ is positive definite, the worst-case Q-vector under a given policy admits the closed-form solution:

$$q^*(s,\cdot;\phi) = \hat{\mu}(s) - \widehat{\Upsilon}(s) \cdot \frac{\widehat{\Sigma}(s)\pi_\phi(\cdot \mid s)}{\|\widehat{\Sigma}(s)^{1/2}\pi_\phi(\cdot \mid s)\|}.$$

This is due to:

$$\begin{aligned}
\min_{q \in \mathscr{U}_{\mathrm{ell}}(\widehat{F}^q_\theta(s))} \mathbb{E}_{a \sim \pi_\phi(\cdot|s)}[q(a)] &= \min_{q:(q-\hat{\mu}(s))^\top \widehat{\Sigma}(s)^{-1}(q-\hat{\mu}(s)) \leq \widehat{\Upsilon}(s)^2} \mathbb{E}_{a \sim \pi_\phi(\cdot|s)}[q(a)] \\
&= \min_{q:(q-\hat{\mu}(s))^\top \widehat{\Sigma}(s)^{-1}(q-\hat{\mu}(s)) \leq \widehat{\Upsilon}(s)^2} \langle \pi_\phi(\cdot \mid s), q \rangle \\
&= \min_{\zeta:\|\zeta\| \leq \widehat{\Upsilon}(s)} \langle \pi_\phi(\cdot \mid s), \hat{\mu}(s) + \widehat{\Sigma}^{1/2}\zeta \rangle \\
&\geq \langle \pi_\phi(\cdot \mid s), \hat{\mu}(s) \rangle - \widehat{\Upsilon}(s)\|\widehat{\Sigma}^{1/2}\pi_\phi(\cdot \mid s)\| \\
&= \left\langle \pi_\phi(\cdot \mid s), \hat{\mu}(s) - \widehat{\Upsilon}(s) \cdot \frac{\widehat{\Sigma}(s)\pi_\phi(\cdot \mid s)}{\|\widehat{\Sigma}(s)^{1/2}\pi_\phi(\cdot \mid s)\|} \right\rangle,
\end{aligned}$$

where we employed Cauchy–Schwartz inequality.

We refer the reader to Algorithm 9 for the pseudocode of the training algorithm based on ellipsoidal uncertainty sets. The algorithm implements the robust Bellman backup and policy update described in Section 3.4.2, leveraging ellipsoidal sets to model epistemic uncertainty. Critic targets are constructed by penalizing the expected Q-value with a Mahalanobis norm term aligned with the current policy, while the actor is optimized to maximize the worst-case return within the ellipsoid. For completeness, pseudocode for the box and convex hull variants is provided in Appendix 3.9.3.

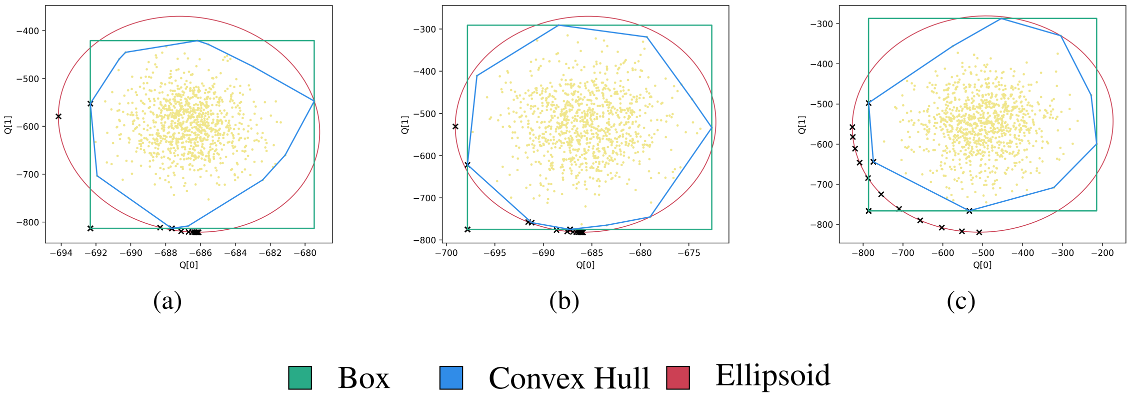
Algorithm 9 : Sample-based Epistemic Robust SAC with Ellipsoidal Uncertainty (ERSAC-E)

```
Input : Initial policy parameters $\phi$, Q parameters $\theta$, target Q parameters $\theta'$,
        offline data replay buffer $\mathscr{D}$, learning rates $\eta_Q, \eta_\pi$, target update rate $\tau$,
        sample size $N$
for each epoch do
    Sample minibatch $\mathscr{B} := \{(s,a,r,s')\}$ from $\mathscr{D}$
    Sample $N$ i.i.d. realizations $\{\tilde{z}_i\}$ from $F_z$
    $\hat{\mu}(s) \leftarrow \frac{1}{N}\sum_{i=1}^{N} \mathfrak{q}_\theta(s,\cdot,\tilde{z}_i)$
    $\widehat{\Sigma}(s) \leftarrow \frac{1}{N}\sum_{i=1}^{N}(\mathfrak{q}_\theta(s,\cdot,\tilde{z}_i) - \hat{\mu}(s))(\mathfrak{q}_\theta(s,\cdot,\tilde{z}_i) - \hat{\mu}(s))^\top$
    $\widehat{\Upsilon}(s) \leftarrow \inf\{\Upsilon |$
        $\frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{(\mathfrak{q}_\theta(s,\cdot,\tilde{z}_i) - \hat{\mu}(s))^\top \widehat{\Sigma}(s)^{-1}(\mathfrak{q}_\theta(s,\cdot,\tilde{z}_i) - \hat{\mu}(s)) \leq \Upsilon^2\} \geq \upsilon\}$
    $\hat{\mu}(s') \leftarrow \frac{1}{N}\sum_{i=1}^{N} \mathfrak{q}_{\theta'}(s',\cdot,\tilde{z}_i)$
    $\widehat{\Sigma}(s') \leftarrow \frac{1}{N}\sum_{i=1}^{N}(\mathfrak{q}_{\theta'}(s',\cdot,\tilde{z}_i) - \hat{\mu}(s'))(\mathfrak{q}_{\theta'}(s',\cdot,\tilde{z}_i) - \hat{\mu}(s'))^\top$
    $\widehat{\Upsilon}(s') \leftarrow \inf\{\Upsilon |$
        $\frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{(\mathfrak{q}_{\theta'}(s',\cdot,\tilde{z}_i) - \hat{\mu}(s'))^\top \widehat{\Sigma}(s')^{-1}(\mathfrak{q}_{\theta'}(s',\cdot,\tilde{z}_i) - \hat{\mu}(s')) \leq \Upsilon^2\} \geq \upsilon\}$
    Compute target:
        $y(r,s') \leftarrow r + \gamma\Big(\langle \pi_\phi(\cdot|s'), \hat{\mu}(s')\rangle - \widehat{\Upsilon}(s')\left\|\widehat{\Sigma}^{1/2}(s')\pi_\phi(\cdot|s')\right\|$
            $- \alpha\,\mathbb{E}_{a'\sim\pi_\phi}\left[\log \pi_\phi(a'|s')\right]\Big)$
    Critic update:
        $\theta \leftarrow \theta - \eta_Q \cdot 2\frac{1}{|\mathscr{B}|}\sum_{(s,a,r,s')\in\mathscr{B}} \mathbb{E}_{\tilde{z}\sim F_z}\left[(\mathfrak{q}_\theta(s,a,\tilde{z}) - y(r,s')) \cdot \nabla_\theta \mathfrak{q}_\theta(s,a,\tilde{z})\right]$
    Actor update:
        $\phi \leftarrow \phi + \eta_\pi \cdot \frac{1}{|\mathscr{B}|}\sum_{s\in\mathscr{B}}\sum_{a\in\mathscr{A}}\left(\hat{\mu}(s,a) - \widehat{\Upsilon}(s)\frac{\widehat{\Sigma}(s)\pi_\phi(\cdot \mid s)}{\left\|\widehat{\Sigma}^{1/2}(s)\pi_\phi(\cdot \mid s)\right\|}\right)\nabla_\phi \pi_\phi(a \mid s)$
            $- \alpha\nabla_\phi \mathbb{E}_{a\sim\pi_\phi(\cdot|s)}\left[\log \pi_\phi(a \mid s)\right]$
    Update target network: $\theta' \leftarrow \tau\theta + (1-\tau)\theta'$
```

3.5.4 Sensitivity of Worst-Case Q Vector to π_ϕ

While the box set yields a fixed $q^*(s,\cdot;\phi)$ independent of the policy, both the convex hull and ellipsoidal sets adapt their minimizer $q^*(s,\cdot;\phi)$ to $\pi_\phi(\cdot \mid s)$. This flexibility introduces a richer learning dynamic, allowing the Bellman backup to respond differently depending on the current policy. From game theoretic point of view, at each state s, the agent proposes a policy $\pi_\phi(\cdot \mid s)$, and an adversary selects the worst-case Q-vector $q^*(s,\cdot;\phi) \in \mathscr{U}_\theta(s)$ that minimizes the expected return $\langle \pi_\phi(\cdot \mid s), q \rangle$. When the uncertainty set contains multiple non-dominated extremal points, as is the case for convex hulls and ellipsoids, the Bellman update becomes more responsive, capable of adjusting its conservativeness based on the agent's action preferences.

To illustrate this, consider the Machine Replacement example discussed in Section 3.1. Figure 3.1 highlights this adaptivity across selected states by comparing the q^* responses of the three sets $\mathscr{U}_{\text{box}}(s), \mathscr{U}_{\text{hull}}(s)$ and $\mathscr{U}_{\text{ell}}(s)$ as the policy π varies uniformly over the probability simplex. This behavior leads to a more expressive training process that is sensitive to the epistemic structure captured by the generative model.



(a)

(b)

(c)

Figure 3.1: (a)–(c): Uncertainty sets and worst-case policy evaluations for states 0, 5, and 10 in the machine replacement example at epoch 1. Each subplot illustrates the distribution of ensemble Q-values along with the corresponding box, convex hull, and ellipsoidal uncertainty sets. Markers "X" indicate the worst-case Q-value q^* under different policies π.

This adaptivity is particularly important in offline settings, where data coverage is often limited or biased. Structured uncertainty sets enable value estimates that are conservative in underexplored regions while remaining responsive in well-covered ones, leading to improved generalization without excessive pessimism.

The construction of these sets connects with the recent evolving literature in Estimate-then-Optimize Conditional Robust Optimization (CRO). One line of work as proposed in Chenreddy, Bandi, and Delage (2022), Goerigk and Kurtz (2020), Ohmori (2021), Sun, Liu, and Li (2023), and Blanquero, Carrizosa, and Gómez-Vargas (2023) focuses on calibrating uncertainty sets over realizations drawn from a conditional distribution $F(q \mid s)$. These methods construct high-probability sets $\mathscr{U}(s) \subset \mathbb{R}^d$ such that for a random realization $q \sim F(\cdot \mid s)$, it holds that $\mathbb{P}(q \in \mathscr{U}(s)) \geq 1 - \delta$. Such calibrated sets enable robust decisions of the form $\max_{\pi \in \Pi} \min_{q \in \mathscr{U}(s)} \pi^\top q$, that ensure performance against probable realizations of the uncertain quantity q, conditioned on covariates s.

A second line of work, common in distributionally robust optimization and robust RL constructs ambiguity sets over the distribution $F(\cdot \mid s)$ itself, e.g., using moment constraints, Wasserstein balls, or scenario-based support (Bertsimas, McCord, and Sturt 2022; C. McCord 2019; Wang and Chen 2020; I. Wang et al. 2023; Nguyen et al. 2021; Esteban-Pérez and Morales 2022). In this setting, one solves:

$$\max_{\pi \in \Pi} \min_{F \in \mathscr{F}(s)} \mathbb{E}_{q \sim F}[\pi^\top q] = \max_{\pi \in \Pi} \min_{\bar{q} \in \mathscr{U}(s)} \pi^\top \bar{q},$$

where $\mathscr{F}(s)$ is an ambiguity set over distributions and $\mathscr{U}(s) := \{\mathbb{E}_{q \sim F}[q] : F \in \mathscr{F}(s)\}$ is the implied uncertainty set over expected values.

Our work aligns more closely with the former, wherein we directly parameterize and sample from a learned conditional distribution $\widehat{F}^q_\theta(s)$, and define a structured uncertainty set $\mathscr{U}(\widehat{F}^q_\theta(s))$ over sampled realizations $q \sim \widehat{F}^q_\theta(s)$. This allows us to reason about epistemic variability in Q-values without requiring a full ambiguity set over $F^q_\theta(s)$. Bridging these two lines of work could lead to rich formulations for epistemically robust reinforcement learning, which we leave for future work.

3.6 The ERSAC Model with Epinet (ERSAC(Epi))

Recall from Assumption 3.4.1 that we require a parametric sampling operator $\mathrm{q}_\theta(s,a,z)$, with $z \sim F_z$, such that $\mathrm{q}_\theta(s,\cdot,z) \sim F^q_\theta(s)$, where $F^q_\theta(s) \in \mathscr{M}(\mathbb{R}^{|\mathscr{A}|})$ denotes a distribution over Q-value vectors. We instantiate this generative model using an Epistemic Neural Network (Epinet) introduced by Osband et al. (2023), which enables structured and differentiable sampling from a single neural network. An Epinet supplements a base network $\mu_{\theta_\mu}(s,a) \in \mathbb{R}$, parameterized by θ_μ, which yields the mean Q-value vector. From this base, we extract a feature representation $\psi_{\theta_\mu}(s) \in \mathbb{R}^{d_\psi}$, typically taken from the last hidden layer. Epistemic variation is introduced via a latent index $z \sim \mathscr{N}(0,I) \in \mathbb{R}^{d_z}$. These components are combined through a stochastic head $\sigma_{\theta_\sigma}(\psi_{\theta_\mu}(s),a,z) \in \mathbb{R}$, which modulates the structured uncertainty. The sampling operator for the Q-value vector is then defined as,

$$\mathrm{q}_\theta(s,\cdot,z) := \mu_{\theta_\mu}(s,\cdot) + \sigma_{\theta_\sigma}(\psi_{\theta_\mu}(s),\cdot,z). \tag{3.20}$$

The stochastic head is constructed as:

$$\sigma_{\theta_\sigma}(\psi,\cdot,z) := \sigma^{\mathrm{L}}_{\theta_\sigma}(\psi,\cdot,z) + \sigma^{\mathrm{P}}(\psi,\cdot,z), \tag{3.21}$$

with $\sigma^{\mathrm{L}}_{\theta_\sigma} : \mathbb{R}^{d_\psi} \times \mathscr{A} \times \mathbb{R}^{d_z} \to \mathbb{R}$ as a learnable function and $\sigma^{\mathrm{P}} : \mathbb{R}^{d_\psi} \times \mathscr{A} \times \mathbb{R}^{d_z} \to \mathbb{R}$ as a fixed prior. The fixed prior network σ^{P} encodes initial epistemic uncertainty by inducing variability in predictions across samples of indices z. In well explored regions, $\sigma^{\mathrm{L}}_{\theta_\sigma}$ can learn better distributions for the predictive uncertainty, while in data sparse areas, σ^{P} can induce the prior beliefs of the decision maker to guide conservative predictions. We can now use it to generate the realizations of the Q-value vectors at a given state s by drawing $z \sim \mathscr{N}(0,I)$ to form the empirical distribution $\widehat{F}_\theta(s)$ over Q values. This enables us to employ the sample based epistemic uncertainty sets introduced earlier in the Section 3.5.

This construction yields a parameter efficient and fully differentiable reparameterization of the Q distribution. Further, one can train these networks using a perturbed squared loss inspired by Gaussian bootstrapping following the loss:

$$\mathscr{L}_Q^{ENN}(\theta) := \mathbb{E}_{(s,a,r,s',c)\sim\bar{\mathscr{D}},\ \tilde{z}\sim F_z}\left[\left(\mathrm{q}_\theta(s,a,z) - y(r,s') - \bar{\sigma}\langle c,z\rangle\right)^2\right] + \lambda_\mu\|\theta_\mu\|^2 + \lambda_\sigma\|\theta_\sigma\|^2, \tag{3.22}$$

where each member (s,a,r,s') from the dataset $\mathscr{D}$ is augmented with some c randomly sampled from the surface of the unit sphere $\mathbb{S}^{d_z}$ to produce $\bar{\mathscr{D}}$, where $\bar{\sigma} > 0$ denotes the bootstrap noise scale, and where $\lambda_\zeta, \lambda_\eta$ are regularization coefficients. This loss encourages the network to match bootstrapped Q-targets while introducing variability across z samples. It can be minimized via standard stochastic gradient methods. The ENN critic updates thus become:

$$\begin{aligned}\theta_\mu \leftarrow \theta_\mu - 2\eta_Q \cdot \Bigg(\frac{1}{|\mathscr{B}|} \sum_{(s,a,r,s',c)\in\bar{\mathscr{B}}} \mathbb{E}_{\tilde{z}\sim F_z}\Big[(\mathrm{q}_\theta(s,a,\tilde{z}) - y(r,s') \\ - \bar{\sigma}\langle c,\tilde{z}\rangle) \cdot \nabla_{\theta_\mu}\mu_{\theta_\mu}(s,a)\Big] + 2\lambda_\mu\theta_\mu\Bigg)\end{aligned} \tag{3.23}$$

$$\begin{aligned}\theta_\sigma \leftarrow \theta_\sigma - 2\eta_Q \cdot \Bigg(\frac{1}{|\mathscr{B}|} \sum_{(s,a,r,s',c)\in\bar{\mathscr{B}}} \mathbb{E}_{\tilde{z}\sim F_z}\Big[(\mathrm{q}_\theta(s,a,\tilde{z}) - y(r,s') \\ - \bar{\sigma}\langle c,\tilde{z}\rangle) \cdot \nabla_{\theta_\sigma}\sigma^L_{\theta_\sigma}(\psi_{\theta_\mu}(s),a,z)\Big] + 2\lambda_\mu\theta_\mu\Bigg)\end{aligned} \tag{3.24}$$

In order to accelerate the evaluation of $\mathscr{U}(F_\theta^q(s)$ when employing an ellipsoidal uncertainty set operator, we introduce additional structure in $\sigma^{\mathrm{L}}_{\theta_\sigma}(\psi,\cdot,z)$ and $\sigma^{\mathrm{P}}(\psi,\cdot,z)$ as outlined in Assumption 3.6.1, namely that both operators are linear in z.

Assumption 3.6.1. *The stochastic heads $\sigma^L_{\theta_\sigma}(\psi,\cdot,z)$ and $\sigma^P(\psi,\cdot,z)$ are linear in z, i.e.*

$$\sigma^L_{\theta_\sigma}(\psi,a,z) = \langle\bar{\sigma}^L_{\theta_\sigma}(\psi,a),z\rangle \qquad , \qquad \sigma^P(\psi,a,z) = \langle\bar{\sigma}^P(\psi,a),z\rangle,$$

for some $\bar{\sigma}^L_{\theta_\sigma} : \mathbb{R}^{d_\psi}\times\mathscr{A}\to\mathbb{R}^{d_z}$ and $\bar{\sigma}^P : \mathbb{R}^{d_\psi}\times\mathscr{A}\to\mathbb{R}^{d_z}$.

Assumption 3.6.1 induces a Gaussian distribution,

$$\mathrm{q}_\theta(s,\cdot,z) \sim \mathscr{N}(\mu_{\theta_\mu}(s), \Sigma_\theta(s)), \tag{3.25}$$

where the covariance is defined as, $[\Sigma_\theta(s)]_{a,a'} := \langle \bar{\sigma}^L_{\theta_\sigma}(\psi_{\theta_\mu}(s),a) + \bar{\sigma}^P(\psi_{\theta_\mu}(s),a), \bar{\sigma}^L_{\theta_\sigma}(\psi_{\theta_\mu}(s),a') + \bar{\sigma}^P(\psi_{\theta_\mu}(s),a')\rangle$. This gives rise to the Epinet based ellipsoidal set:

$$\mathscr{U}^{ENN}_{\mathrm{ell}}(s) = \left\{ q \in \mathbb{R}^{|\mathscr{A}|} : (q - \mu_{\theta_\mu}(s))^\top \Sigma_\theta(s)^{-1}(q - \mu_{\theta_\mu}(s)) \leq F^{-1}_{\chi^2_{|\mathscr{A}|}}(\upsilon) \right\}, \tag{3.26}$$

where $F^{-1}_{\chi^2_{|\mathscr{A}|}}(\upsilon)$ is the inverse cumulative distribution function of the χ^2 distribution with $|\mathscr{A}|$ degrees of freedom. This construction provides a computationally efficient alternative to ensemble-based epistemic modeling and enables closed-form worst-case Q-vector computation for robust policy evaluation.

The training procedure for ERSAC with Epinet (ERSAC(Epi)) mirrors the structure of the ensemble based approach described in Algorithm 9. However, rather than computing the empirical mean and covariance from sampled Q-values, the special structured Epinet model provides these quantities in closed form. Specifically, the mean vector is given by the deterministic head $\mu_{\theta_\mu}(s)$, and the covariance matrix $\Sigma_\theta(s)$ is derived analytically from the structure of the stochastic head under Assumption 3.6.1. The ellipsoidal radius is set to $\Upsilon^2(s) = F^{-1}_{\chi^2_{|\mathscr{A}|}}(\upsilon)$, corresponding to a υ-confidence level. This eliminates the need for sampling when constructing the uncertainty set, allowing efficient and differentiable computation of the Bellman target and policy gradient. The full ERSAC(Epi) algorithm follows the similiar steps as Algorithm 9 and is deferred to Appendix 3.9.3 for completeness.

3.7 Experiments

In this section, we present a comprehensive empirical evaluation of our proposed framework for epistemic robustness in offline reinforcement learning. We quantify epistemic uncertainty through uncertainty sets that can be seamlessly integrated into robust policy optimization. In Section 3.5, we introduced three types of sample-based uncertainty sets: box, convex hull, and ellipsoid constructed from distributions over Q-values. Based on these constructions, we instantiate three corresponding methods within the ERSAC framework: **ERSAC-B-N**, which uses box set constructed over N ensembles, **ERSAC-CH-N**, which employs convex hulls over the N ensembles and **ERSAC-Ell-N**, which forms ellipsoidal sets using the empirical mean and covariance of the ensemble. We further introduce the **ERSAC-Ell-Epi** model from Section 3.6, which replaces the ensemble with samples drawn from an Epistemic Neural Network (Epinet). To maintain consistency with the ensemble-based variants, we sample N latent indices z from the Epinet to generate Q-value vectors. Finally, we propose **ERSAC-Ell-Epi*** which leverages the special structure for the stochastic head $\sigma_{\theta_\sigma}(\psi, \cdot, z)$ as defined in Assumption 3.6.1 to constructs the ellipsoidal set directly without requiring sampling of Q vectors. We benchmark these methods against the standard **SAC-N** baseline, which we have shown as a special case of our framework under a box uncertainty set. To remain consistent with prior literature, we will refer to **ERSAC-B-N** as **SAC-N** throughout the experimental section.

Our experiments span a diverse set of environments, including tabular domains (Machine Replacement and Riverswim), classic control benchmarks (CartPole and LunarLander). Across these domains, we evaluate the ability of each method to learn effective policies under distributional shifts arising due to changes in the behavioral policies generating the data and limited data coverage.

A key contribution of our experimental setup is a novel offline RL benchmarking framework that enables control over the risk sensitivity of the behavior policy used to generate offline datasets. By adjusting the level of optimism or pessimism through expectile-based value learning, we can systematically evaluate how the nature of behavioral data affects the performance of offline RL algorithms.

To induce risk sensitivity in behavior policies, we adopt a modified actor-critic algorithm that incorporates the dynamic expectile risk measure (Marzban, Delage, and Li 2023). This implementation constructs one-step expectile targets using a bootstrapped Bellman update. Specifically, for each traversed (s,a), we compute the target as

$$\begin{aligned} y := \sup\Big\{ z : \\ &\mathbb{E}_{s' \sim \hat{p}_{N_s}(\cdot|s,a)} \Big[\Big| \tau - \mathbb{I}\Big(z < r + \gamma \max_{a'} Q_{\theta'}(s',a') \Big) \Big| \cdot \Big(z - r - \gamma \max_{a'} Q_{\theta'}(s',a') \Big) \Big] \leq 0 \Big\}, \end{aligned}$$

where $\hat{p}_{N_s}(\cdot|s,a)$ is the empirical distribution of N_s resampling of the transition from (s,a).

We then update the critic to minimize the squared error to this expectile target. The actor is trained using standard policy gradient objective, which seeks to maximize the expected Q values. After a fixed number of training episodes, the resulting policy π_ϕ reflects the desired level of risk sensitivity encoded by τ. We then use this policy to collect an offline dataset of size $N_{\mathscr{D}}$ via ε-greedy interaction with the environment, selecting a random action with probability $\varepsilon = 0.1$. This process yields datasets that systematically vary in their underlying behavioral bias. Full details of the implementation are provided n Appendix 3.9.3 (see Algorithm 13).

3.7.1 Evaluation on Tabular Tasks

We begin our evaluation with focusing on two popular tabular MDP environments, *Machine Replacement* problem and *Riverswim*. These settings offer interpretable structure while capturing key challenges in offline RL, including sparse state-action coverage, and high sensitivity to policy extrapolation. More importantly, the tabular setup allows us to isolate the effects of epistemic uncertainty arising from limited data coverage, without confounding factors introduced by function approximators used in deep RL such as overfitting, instability, or extrapolation error. This enables a clean evaluation of how different uncertainty set constructions mitigate overestimation in offline learning, specifically in settings where epistemic uncertainty is the dominant source of error. For each environment, we construct a variety of offline datasets by systematically varying two key parameters, dataset size and behavior policy risk sensitivity. To evaluate sample efficiency, we vary the dataset size across three levels, $10 \times |\mathscr{S}|$, $100 \times |\mathscr{S}|$, and $1,000 \times |\mathscr{S}|$, where $|\mathscr{S}|$ denotes the number of states in the environment. These correspond to increasing levels of coverage over the state-action space and allow us to systematically study the impact of data availability on policy performance. Empirically, we observe that beyond $1,000 \times |\mathscr{S}|$ samples, the learned empirical transition dynamics closely approximate the true transition model, yielding diminishing returns from additional data. To induce behavioral bias and control epistemic uncertainty, we vary the behavior policy using the dynamic expectile risk measure at three levels: risk-seeking ($\tau = 0.1$), risk-neutral ($\tau = 0.5$), and risk-averse ($\tau = 0.9$). These settings correspond to qualitatively distinct exploration profiles and result in datasets with varying coverage of the state-action space.

We evaluate performance using normalized returns, which measure the improvement of a learned policy over a uniformly random policy, scaled relative to the performance of the optimal policy. Specifically, for a learned policy π, we compare

$$(J(\pi) - J(\pi_{\text{rand}})) \,/\, (J(\pi^*) - J(\pi_{\text{rand}}))$$

where $J(\pi)$ is the expected returns under policy π, computed as the average return over 100 evaluation episodes, π_{rand} is the random policy, and π^* is the optimal policy.

Table 3.1 summarizes normalized returns across methods and data regimes. We observe that in small datasets (e.g., 100 samples), CH-N and Ell_0.9-N outperform B-N by up to 75% demonstrating the advantage of structured epistemic reasoning in the case of low coverage. As dataset size increases, all methods improve, but structured uncertainty sets tend to converge more quickly toward optimal returns. Under risk-averse data regimes ($\tau = 0.9$), where epistemic uncertainty is highest, ellipsoidal variants remain robust, with Ell-N and Ell_0.9-N effectively modulating conservativeness to sustain performance.

A key advantage of the ellipsoidal uncertainty set is its tunable scaling parameter ε, which controls the conservativeness of the set. To validate its impact, we compare ellipsoids constructed to cover 100% of ensemble samples (Ell-N) versus 90% (Ell_0.9-N). Empirically, we observe that the more compact ellipsoid with 90% coverage often yields better performance, likely due to excluding outlier critics and avoiding excessive pessimism. Based on this finding, we adopt the 90% coverage threshold as the default configuration for ellipsoidal sets in subsequent Gym based experiments.

3.7.2 Evaluation on Gym Environments

We next evaluate the proposed methods on two widely used Gym environments—*CartPole* and *LunarLander*. CartPole is a well known control problem involving binary rewards and continuous states, while LunarLander presents a more complex challenge with continuous

Env	DS	τ	SAC-N	CH-N	Ell-N	Ell_0.9-N	Beh. Policy
MR	10×	0.1	80 ± 3	85 ± 2	87 ± 1	$\mathbf{88 \pm 2}$	86 ± 3
	100×	0.1	97 ± 1	97 ± 1	95 ± 2	96 ± 2	86 ± 3
	1,000×	0.1	98 ± 2	98 ± 2	96 ± 2	96 ± 1	86 ± 3
	10×	0.5	87 ± 2	88 ± 2	90 ± 2	$\mathbf{91 \pm 2}$	100 ± 0
	100×	0.5	97 ± 1	$\mathbf{98 \pm 1}$	92 ± 2	94 ± 2	100 ± 0
	1,000×	0.5	98 ± 2	98 ± 2	98 ± 2	$\mathbf{99 \pm 0}$	100 ± 0
	10×	0.9	85 ± 2	86 ± 2	90 ± 2	90 ± 2	92 ± 2
	100×	0.9	96 ± 2	96 ± 2	95 ± 2	96 ± 2	92 ± 2
	1,000×	0.9	96 ± 2	96 ± 2	96 ± 2	96 ± 1	92 ± 2
RS	10×	0.1	37 ± 4	64 ± 2	57 ± 3	$\mathbf{66 \pm 3}$	-20 ± 3
	100×	0.1	92 ± 2	94 ± 2	94 ± 3	94 ± 3	-20 ± 3
	1,000×	0.1	99 ± 1	100 ± 0	100 ± 0	100 ± 0	-20 ± 3
	10×	0.5	56 ± 2	60 ± 2	60 ± 2	$\mathbf{62 \pm 1}$	100 ± 0
	100×	0.5	97 ± 2	99 ± 1	98 ± 1	99 ± 1	100 ± 0
	1,000×	0.5	99 ± 1	99 ± 1	100 ± 0	100 ± 0	100 ± 0
	10×	0.9	49 ± 2	49 ± 4	48 ± 1	$\mathbf{52 \pm 3}$	34 ± 4
	100×	0.9	99 ± 1	99 ± 1	$\mathbf{100 \pm 0}$	99 ± 1	34 ± 4
	1,000×	0.9	99 ± 1	99 ± 1	100 ± 0	100 ± 0	34 ± 4

Table 3.1: Normalized returns with 90% confidence interval achieved by SAC-N, CH-N, Ell-N, and Ell_0.9-N across dataset sizes $\{10\times, 100\times, 1,000\times\}$ and behavior policy risk levels $\tau \in \{0.1, 0.5, 0.9\}$ in the Machine Replacement and RiverSwim environments. Scores are computed over 10 evaluation seeds and normalized relative to the random and optimal policy baselines. Bold and underline highlight respectively the best and worst performing method when the margin is larger or equal to one. The final column reports the return of the behavior policy used to generate the offline data.

states, shaped rewards, and a higher dimensional state-action space. Similiar to the tabular setting, To construct the offline datasets, we again vary two key factors: dataset size and behavior policy risk profile. For each environment, we generate nine datasets by crossing three data sizes, $1,000$, $10,000$, and $100,000$ transitions and with three expectile levels, $\tau = 0.1$ (risk-seeking), $\tau = 0.5$ (risk-neutral), and $\tau = 0.9$ (risk-averse). Behavior policies are trained to convergence using a dynamic expectile based actor-critic model, and fixed trajectories are collected for each configuration.

Table 3.2 presents normalized returns across the different methods and dataset regimes.

Env	**DS**	τ	**SAC-N**	**CH-N**	**Ell_0.9-N**	**Ell-Epi**	**Ell-Epi***	**Beh. Policy**
CP	1k	0.1	84 ± 3	$\underline{81 \pm 2}$	$\mathbf{86 \pm 1}$	84 ± 1	85 ± 2	86 ± 2
	10k	0.1	$\underline{92 \pm 2}$	94 ± 2	100 ± 0	100 ± 0	100 ± 0	86 ± 2
	100k	0.1	100 ± 0	100 ± 0	100 ± 0	100 ± 0	100 ± 0	86 ± 2
	1k	0.5	$\underline{70 \pm 2}$	72 ± 1	$\mathbf{73 \pm 3}$	72 ± 2	71 ± 2	100 ± 0
	10k	0.5	$\underline{97 \pm 2}$	99 ± 1	100 ± 0	100 ± 0	100 ± 0	100 ± 0
	100k	0.5	100 ± 0	100 ± 0	100 ± 0	100 ± 0	100 ± 0	100 ± 0
	1k	0.9	73 ± 2	$\underline{70 \pm 3}$	78 ± 2	$\mathbf{80 \pm 1}$	75 ± 2	83 ± 2
	10k	0.9	100 ± 0	100 ± 0	100 ± 0	100 ± 0	100 ± 0	83 ± 2
	100k	0.9	100 ± 0	100 ± 0	100 ± 0	100 ± 0	100 ± 0	83 ± 2
LL	1k	0.1	$\underline{72 \pm 1}$	77 ± 1	98 ± 2	97 ± 3	98 ± 2	94 ± 3
	10k	0.1	$\underline{94 \pm 2}$	98 ± 1	102 ± 1	102 ± 3	$\mathbf{103 \pm 1}$	94 ± 2
	100k	0.1	$\underline{99 \pm 1}$	100 ± 3	106 ± 1	$\mathbf{110 \pm 3}$	108 ± 1	94 ± 2
	1k	0.5	$\underline{68 \pm 3}$	73 ± 3	96 ± 3	95 ± 1	$\mathbf{97 \pm 1}$	100 ± 2
	10k	0.5	$\underline{93 \pm 3}$	99 ± 1	100 ± 1	99 ± 1	$\mathbf{102 \pm 1}$	100 ± 2
	100k	0.5	$\underline{98 \pm 2}$	100 ± 1	102 ± 2	$\mathbf{108 \pm 2}$	105 ± 2	100 ± 2
	1k	0.9	$\underline{67 \pm 2}$	73 ± 2	97 ± 2	$\mathbf{98 \pm 2}$	97 ± 2	78 ± 3
	10k	0.9	92 ± 2	$\underline{92 \pm 3}$	101 ± 2	100 ± 4	$\mathbf{102 \pm 2}$	78 ± 3
	100k	0.9	$\underline{98 \pm 2}$	101 ± 2	103 ± 1	104 ± 2	$\mathbf{105 \pm 1}$	78 ± 3

Table 3.2: Normalized returns with 90% confidence intervals achieved by the five algorithms across dataset sizes $\{1\text{k}, 10\text{k}, 100\text{k}\}$ and behavior-policy risk levels $\tau \in \{0.1, 0.5, 0.9\}$ in CartPole and LunarLander. Scores are averaged over 10 evaluation seeds and normalized against random and optimal baselines. Bold and underline highlight respectively the best and worst performing method when the margin is larger or equal to one.

We consider the policy trained under the risk neutral behavior($\tau = 0.5$) as the reference optimal policy. First, models CH-N, Ell_0.9-N, Ell-Epi consistently outperform the box baseline B-N, particularly in data scarce and risk averse settings where epistemic uncertainty plays a larger role. When we aggregate returns across dataset sizes by risk level (As presented in Table 3.3), we observe that Ell_0.9-N consistently achieves strong performance under risk-neutral and risk-seeking behavior policies, suggesting that the method effectively leverages optimistic data to enhance policy learning.

Further, ellipsoidal variants offer robust and often best performance across most settings. Notably, Ell-Epi* matches or surpasses ensemble based Ell_0.9-N in several

Env	$\tau = 0.1$	$\tau = 0.5$	$\tau = 0.9$
MR	93 ± 5	**95 ± 4**	94 ± 3
RS	87 ± 15	87 ± 17	84 ± 22
CartPole	**95 ± 8**	93 ± 14	92 ± 14
LunarLander	**103 ± 7**	99 ± 5	99 ± 5

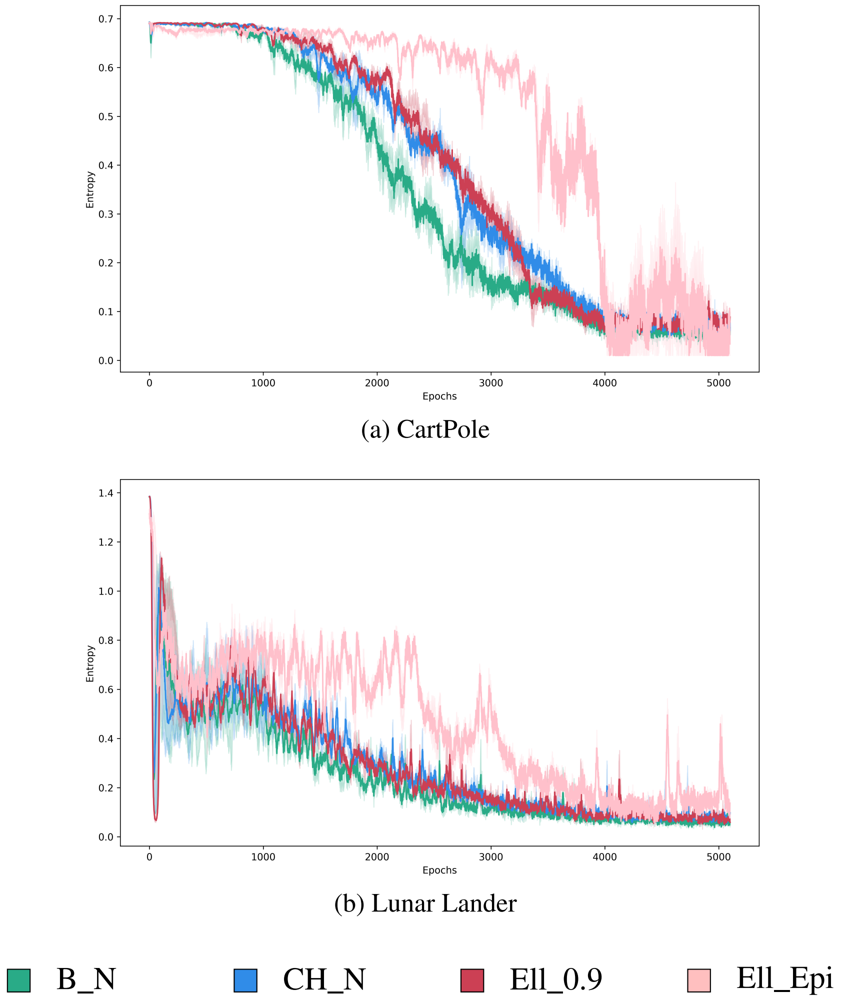
Table 3.3: Aggregated performance of Ell_0.9-N across environments with mean ± standard deviation. Bold and underline highlight respectively the best and worst performing source of data.

settings, indicating that Epinet based uncertainty representations can serve as lightweight and effective alternatives to ensembles. This advantage is further supported by runtime measurements (Table 3.4), which show that the Epinet based model, which learns the ellipsoidal sets directly, are able to achieve comparable performance with significantly lower computational cost, making it an attractive choice for scaling to more complex domains.

Model	**SAC-N**	**CH-N**	**Ell_0.9-N**	**Ell-Epi**	**Ell-Epi***
Runtime (s/epoch)	0.35	0.42	0.56	0.60	0.10

Table 3.4: Runtime per training epoch for each model in LunarLander with 100,000 offline transitions and $\tau = 0.5$, averaged over 10 seeds

To further understand how uncertainty sets affect learning dynamics, we analyze policy entropy during training. Figure 3.2 shows that Box-based methods (B-N) exhibit consistently lower entropy throughout training, indicating less stochastic policies. This behavior leads to early convergence toward deterministic actions, which may result in suboptimal local solutions. While all methods eventually stabilize, as discussed in Section 3.5.4, **CH-N**, **Ell-N**, and **Ell-Epi** offer greater flexibility in shaping the value function $q^*(s, \cdot; \phi)$. This flexibility translates to more exploratory behavior when deriving the policy $\pi_\phi(\cdot \mid s)$, ultimately enabling better identification of high-performing actions under offline constraints.



(a) CartPole

(b) Lunar Lander

Figure 3.2: Policy entropy during training across B_N, CH_N, Ell_0.9, and Ell_Epi models in the *CartPole* and *LunarLander* environments. Entropy is computed per epoch and averaged over 10 evaluation seeds. Lower entropy indicates more confident, deterministic policies, while higher entropy reflects greater stochasticity.

3.8 Conclusion

This chapter presented Epistemic Robust Soft Actor-Critic (ERSAC), a unified framework for offline reinforcement learning that robustly accounts for epistemic uncertainty through structured uncertainty sets over Q-values. By replacing traditional ensemble based pessimism with compact and expressive uncertainty sets, ERSAC enables conservative yet flexible value estimation and policy optimization. We showed that our framework generalizes SAC-N as a special case, and supports multiple set constructions such as box, convex hull, and ellipsoids, each offering trade-offs in expressiveness and computational cost. Building on this, we introduced an Epinet based variant of ERSAC that generates ellipsoidal uncertainty sets in closed form, thus eliminating the need for sample-based approximations and significantly reducing runtime without compromising performance.

Through comprehensive evaluations across tabular and continuous environments, we demonstrated that ERSAC variants, particularly those using ellipsoidal and Epinet-based sets, achieve better performance. Our experiments also introduced a novel benchmark to systematically assess offline RL algorithms under varying degrees of risk sensitivity in behavior policies, highlighting the importance of aligning epistemic modeling with the data generation process.

Beyond these results, ERSAC demonstrates how set-based modeling can replace variance inflation or large ensembles as a principled approach to epistemic uncertainty. By explicitly bounding plausible Q-values, the Bellman backup adapts its conservativeness to the data by being nearly deterministic in well covered states, and cautious in underexplored ones. Such adaptivity is especially valuable in offline to online deployment and sim-to-real transfer. In sim-to-real settings, the gap between simulated dynamics and real world behavior often leads to systematic performance drops. Existing approaches typically address this mismatch through domain randomization or by inflating ensemble variance, both of which can be computationally expensive and prone to over conservatism. ERSAC instead frames these discrepancies as a form of epistemic uncertainty, modeling them with structured sets that explicitly capture the range of plausible Q-values. By doing so, ERSAC ensures that policies remain cautious in regions where the simulator is unreliable, while still exploiting reliable aspects of the model without excessive conservatism.

There remain, however, several promising directions for future research. One natural extension is to construct uncertainty sets that are robust not only to epistemic variation but also to distributional ambiguity, thereby capturing a broader range of model misspecification. Another direction involves incorporating risk-sensitive objectives directly into the learning process so that agents can explicitly account for tail events in returns. Extending epistemic robustness to multi-agent and hierarchical reinforcement learning is also compelling, as it would require coordinating uncertainty across interacting agents or abstraction layers. Furthermore, many real-world environments feature state-dependent feasible actions, $\mathscr{A}(s) \subseteq \mathscr{A}$, rather than a single global action set. In such cases the uncertainty set $\mathscr{U}_\theta(s)$ must be restricted to feasible coordinates, altering its geometry and influencing the adversary's choice of $q^*(s,\cdot;\phi)$. Finally, while our methods show strong empirical performance, establishing finite-sample guarantees and deriving robust regret bounds under epistemic uncertainty remain important theoretical challenges. Taken together, these contributions highlight that structured and computationally efficient epistemic modeling offers a foundation for safe, generalizable, and scalable offline reinforcement learning.

Acknowledgements

The authors thank Esther Derman and Wenjie Huang for helpful discussions. The authors also gratefully acknowledge support from the Institut de Valorisation des Données (IVADO), the Canadian Natural Sciences and Engineering Research Council [RGPIN-2022-05261], and the Canada Research Chair program [CRC-2018-00105].

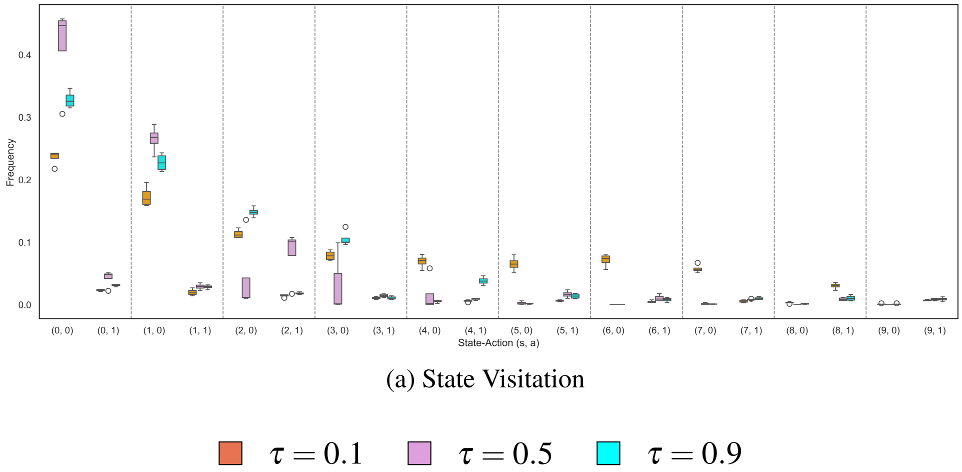
3.9 Appendix

This appendix provides theoretical and implementation details that support our main results. Section 3.9.2 presents a formal lemma and proofs. Section 3.9.3 contains algorithmic pseudocode for the ERSAC variants proposed in this work and Section 3.9.3 details the details regarding experiments training and additional analysis.

3.9.1 Machine Replacement Example

τ	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
0.1	0	0	0	0	0	0	0	0	1	1
0.5	0	0	0	0	0	0	1	1	1	1
0.9	0	0	0	0	0	1	1	1	1	1

Table 3.5: Optimal actions for each state under different expectile levels τ. Action 0 corresponds to progressing forward; Action 1 corresponds to jumping to state 1 with -100 reward.



(a) State Visitation

Figure 3.3: State visitation frequency distributions under different expectile policies.

3.9.2 Proof for Proposition 3.4.1

We begin by analyzing the robust estimator term present in both the conservative target value in equation (3.10) and the policy loss in (3.12): $\min_{q\in\mathscr{U}_\theta(s)}\langle\pi_\phi(\cdot\mid s),q\rangle$. Given that the uncertainty set is defined as a coordinate-wise product box and that $\pi_\phi(\cdot \mid s) \geq 0$, the minimum must be achieved at the coordinate-wise lower bound:

$$q^*(a) = \mathrm{essinf}_{\tilde{q} \sim F_\theta^q(s)}[\tilde{q}(a)] = \mathrm{essinf}_{\tilde{i} \sim U(N)}[Q_{\theta_{\tilde{i}}}(s,a)] = \min_{i \in [N]} Q_{\theta_i}(s,a), \quad \forall a \in \mathscr{A}.$$

The robust evaluation then becomes,

$$\min_{q \in \mathscr{U}_\theta(s)} \langle \pi_\phi(\cdot \mid s), q \rangle = \sum_{a \in \mathscr{A}} \pi_\phi(a \mid s) \min_{i \in [N]} Q_{\theta_i}(s,a) = \mathbb{E}_{a \sim \pi_\phi(\cdot|s)} \left[\min_{i \in [N]} Q_{\theta_i}(s,a) \right].$$

Hence, the conservative target value becomes

$$\begin{aligned} y(r,s') &= r + \gamma \left(\mathbb{E}_{a' \sim \pi_\phi(\cdot|s')} \left[\min_{i \in [N]} Q_{\theta_i}(s',a') - \alpha \log \pi_\phi(a' \mid s') \right] \right) \\ &= \mathbb{E}_{a' \sim \pi_\phi(\cdot|s')} \left[r + \gamma \left(\min_{i \in [N]} Q_i(s',a') - \alpha \log \pi_\phi(a' \mid s') \right) \right] \\ &= \mathbb{E}_{a' \sim \pi_\phi(\cdot|s')} \left[y(r,s',a') \right] \end{aligned}$$

We thus have that

$$\begin{aligned} \mathscr{L}_Q^R(\theta) &= \mathbb{E}_{(s,a,r,s') \sim \mathscr{D},\ \tilde{q} \sim F_\theta^q(s)} \left[\left(\tilde{q}(a) - y(r,s') \right)^2 \right] \\ &= \mathbb{E}_{(s,a,r,s') \sim \mathscr{D},\ \tilde{q} \sim F_\theta^q(s)} \left[\left(\tilde{q}(a) - \mathbb{E}_{a' \sim \pi_\phi(\cdot|s')}[y(r,s',a')] \right)^2 \right] \\ &= \mathbb{E}_{(s,a,r,s') \sim \mathscr{D},\ \tilde{q} \sim F_\theta^q(s)} \left[\tilde{q}(a)^2 - 2\tilde{q}(a) \mathbb{E}_{a' \sim \pi_\phi(\cdot|s')}[y(r,s',a')] + \mathbb{E}_{a' \sim \pi_\phi(\cdot|s')}[y(r,s',a')]^2 \right] \\ &= \mathbb{E}_{(s,a,r,s') \sim \mathscr{D},\ \tilde{q} \sim F_\theta^q(s)} \left[\tilde{q}(a)^2 - 2\tilde{q}(a) \mathbb{E}_{a' \sim \pi_\phi(\cdot|s')}[y(r,s',a')] + \mathbb{E}_{a' \sim \pi_\phi(\cdot|s')}[y(r,s',a')^2] \right] \\ &\quad + \mathbb{E}_{(s,a,r,s') \sim \mathscr{D}} \left[\mathbb{E}_{a' \sim \pi_\phi(\cdot|s')}[y(r,s',a')]^2 - \mathbb{E}_{a' \sim \pi_\phi(\cdot|s')}[y(r,s',a')^2] \right] \\ &= \mathbb{E}_{(s,a,r,s') \sim \mathscr{D},\ \tilde{q} \sim F_\theta^q(s), a' \sim \pi_\phi(\cdot|s')} \left[\tilde{q}(a)^2 - 2\tilde{q}(a) y(r,s',a') + y(r,s',a')^2 \right] + C \\ &= \mathbb{E}_{(s,a,r,s') \sim \mathscr{D},\ \tilde{q} \sim F_\theta^q(s), a' \sim \pi_\phi(\cdot|s')} \left[\left(\tilde{q}(a) - y(r,s',a') \right)^2 \right] + C \\ &= (1/N) \sum_i \mathbb{E}_{(s,a,r,s') \sim \mathscr{D}, a' \sim \pi_\phi(\cdot|s')} \left[\left(Q_{\theta_i}(s,a) - y(r,s',a') \right)^2 \right] + C \\ &= (1/N) \sum_i \mathscr{L}_Q(\theta_i) + C \end{aligned}$$

where

$$C := \mathbb{E}_{(s,a,r,s') \sim \mathscr{D}}[(\mathbb{E}_{a' \sim \pi_\phi(\cdot|s')}[y(r,s',a')])^2] - \mathbb{E}_{(s,a,r,s') \sim \mathscr{D}, a' \sim \pi_\phi(\cdot|s')} \left[y(r,s',a')^2 \right]$$

due to $\tilde{q}(a)$ being independent of $y(r, s', a')$ given (s, a, r, s').

On the other hand, we have that:

$$\begin{aligned}
\mathscr{J}_\pi^R(\phi) &= \mathbb{E}_{s\sim\mathscr{D}, a\sim\pi_\phi(\cdot|s)}\left[\min_{q\in\mathscr{U}_\theta(s)} \langle \pi_\phi(\cdot \mid s), q\rangle - \alpha \log \pi_\phi(a \mid s)\right] \\
&= \mathbb{E}_{s\sim\mathscr{D}, a\sim\pi_\phi(\cdot|s)}\left[\mathbb{E}_{a'\sim\pi_\phi(\cdot|s)}[\min_{i\in[N]} Q_{\theta_i}(s, a')] - \alpha \log \pi_\phi(a \mid s)\right] \\
&= \mathbb{E}_{s\sim\mathscr{D}, a\sim\pi_\phi(\cdot|s)}\left[\min_{i\in[N]} Q_{\theta_i}(s, a) - \alpha \log \pi_\phi(a \mid s)\right] \\
&= \mathscr{J}_\pi(\phi).
\end{aligned}$$

This completes our proof.

3.9.3 Algorithmic Implementation Details

In this section, we present the pseudo-code for the algorithms discussed in the main work.

ERSAC with Box and Convex Hull Sets

Algorithm 10 : Sample-based Epistemic Robust SAC with Box Set

```
Input : Initial policy parameters $\phi$, Q parameters $\theta$, target Q parameters $\theta'$, offline
        data buffer $\mathscr{D}$, learning rates $\eta_Q, \eta_\pi$, target update rate $\tau$, sample size $N$
for each epoch do
    Sample minibatch $\mathscr{B} := \{(s,a,r,s')\}$ from $\mathscr{D}$
    Sample $N$ i.i.d. latent variables $\{\tilde{z}_i\}_{i=1}^{N}$ from $F_z$
    Compute robust targets:
        $$y_{\text{box}}(r,s') := r + \gamma \left( \sum_{a \in \mathscr{A}} \pi_\phi(a|s') \cdot \min_{i \in [N]} \mathrm{q}_{\theta'}(s',a,\tilde{z}_i) - \alpha \sum_{a \in \mathscr{A}} \pi_\phi(a|s') \log \pi_\phi(a|s') \right)$$
    Update critic network:
        $$\theta \leftarrow \theta - \eta_Q \cdot \frac{2}{|\mathscr{B}|} \sum_{(s,a,r,s') \in \mathscr{B}} \mathbb{E}_{\tilde{z} \sim F_z} \left[ (\mathrm{q}_\theta(s,a,\tilde{z}) - y(r,s')) \cdot \nabla_\theta \mathrm{q}_\theta(s,a,\tilde{z}) \right]$$
    Update actor network:
        $$\phi \leftarrow \phi + \eta_\pi \cdot \frac{1}{|\mathscr{B}|} \sum_{s \in \mathscr{B}} \sum_{a \in \mathscr{A}} \min_{i \in [N]} \mathrm{q}_\theta(s,a,\tilde{z}_i) \nabla_\phi \pi_\phi(a|s) - \alpha \nabla_\phi \mathbb{E}_{a \sim \pi_\phi(\cdot|s)} \left[ \log \pi_\phi(a \mid s) \right]$$
    Update target network: $\theta' \leftarrow \tau\theta + (1-\tau)\theta'$
```

Algorithm 11 : Sample-based Epistemic Robust SAC with Convex Hull Set

```
Input : Initial policy parameters $\phi$, Q parameters $\theta$, target Q parameters $\theta'$, offline
        data buffer $\mathscr{D}$, learning rates $\eta_Q, \eta_\pi$, target update rate $\tau$, sample size $N$
for each epoch do
    Sample minibatch $\mathscr{B} := \{(s,a,r,s')\}$ from $\mathscr{D}$
    Sample $N$ i.i.d. latent variables $\{\tilde{z}_i\}_{i=1}^N$ from $F_z$
    Compute robust targets:
        $$y_{\text{hull}}(r,s') := r + \gamma \left( \min_{i\in[N]} \sum_{a\in\mathscr{A}} \pi_\phi(a|s') \cdot \mathsf{q}_{\theta'}(s',a,\tilde{z}_i) - \alpha \sum_{a\in\mathscr{A}} \pi_\phi(a|s') \log \pi_\phi(a|s') \right)$$
    Update critic network:
        $$\theta \leftarrow \theta - \eta_Q \cdot \frac{2}{|\mathscr{B}|} \sum_{(s,a,r,s')\in\mathscr{B}} \mathbb{E}_{\tilde{z}\sim F_z} \left[ (\mathsf{q}_\theta(s,a,\tilde{z}) - y(r,s')) \cdot \nabla_\theta \mathsf{q}_\theta(s,a,\tilde{z}) \right]$$
    Update actor network:
        $$i^*(s,a) := \arg\min_{i\in[N]} \sum_{a\in\mathscr{A}} \pi_\phi(a \mid s) \cdot \mathsf{q}_\theta(s,a,\tilde{z}_i)$$
        $$\phi \leftarrow \phi + \eta_\pi \cdot \frac{1}{|\mathscr{B}|} \sum_{s\in\mathscr{B}} \sum_{a\in\mathscr{A}} \mathsf{q}_\theta(s,a,\tilde{z}_{i^*(s,a)}) \nabla_\phi \pi_\phi(a|s) - \alpha \nabla_\phi \mathbb{E}_{a\sim\pi_\phi(\cdot|s)} \left[ \log \pi_\phi(a \mid s) \right]$$
    Update target network: $\theta' \leftarrow \tau\theta + (1-\tau)\theta'$
```

Epinet-based ERSAC with Ellipsoidal Uncertainty

Algorithm 12 : Sample-based ERSAC with Ellipsoidal Uncertainty using Epinet

```
Input : Initial parameters for policy $\phi$, Q-network $(\theta_\mu, \theta_\sigma)$, and target network
        $(\theta'_\mu, \theta'_\sigma)$; offline data $\mathscr{D}$; learning rates $\eta_Q, \eta_\pi$, and $\tau$; noise scale $\bar{\sigma}$;
        regularization coefficients $\lambda_\mu, \lambda_\sigma$; sample size $N$
for each epoch do
    Sample minibatch $\bar{\mathscr{B}} := \{(s,a,r,s',c)\}$ from augmented buffer $\bar{\mathscr{D}}$
    Sample $N$ i.i.d. latent indices $\{\tilde{z}_i\}_{i=1}^N \sim \mathscr{N}(0,I)$
    Construct uncertainty set (Epinet-based ellipsoid):
        $\hat{\mu}(s') \leftarrow \mu_{\theta'_\mu}(s'), \qquad \bar{\sigma}_{\theta'}(s',a) \leftarrow \bar{\sigma}^L_{\theta'_\sigma}(\psi_{\theta'_\mu}(s'),a) + \bar{\sigma}^P(\psi_{\theta'_\mu}(s'),a)$
        $\Sigma_{\theta'}(s')_{a,a'} \leftarrow \langle \bar{\sigma}_{\theta'}(s',a), \bar{\sigma}_{\theta'}(s',a') \rangle$
    Compute robust targets:
        $y(r,s') \leftarrow r + \gamma \Big( \langle \pi_\phi(\cdot|s'), \hat{\mu}(s') \rangle - \rho \left\| \Sigma^{1/2}_{\theta'}(s') \pi_\phi(\cdot|s') \right\|_2 - \alpha \, \mathbb{E}_{a' \sim \pi_\phi}[\log \pi_\phi(a'|s')] \Big)$
    Update critic network:
        $\theta_\mu \leftarrow \theta_\mu - 2\eta_Q \cdot \frac{1}{|\bar{\mathscr{B}}|} \sum_{(s,a,r,s',c) \in \bar{\mathscr{B}}} \mathbb{E}_{\tilde{z} \sim \mathscr{N}(0,I)} \Big[$
            $\big( \mathfrak{q}_\theta(s,a,\tilde{z}) - y(r,s') - \bar{\sigma} \langle c, \tilde{z} \rangle \big) \cdot \nabla_{\theta_\mu} \mu_{\theta_\mu}(s,a) \Big] + 2\lambda_\mu \theta_\mu$

        $\theta_\sigma \leftarrow \theta_\sigma - 2\eta_Q \cdot \frac{1}{|\bar{\mathscr{B}}|} \sum_{(s,a,r,s',c) \in \bar{\mathscr{B}}} \mathbb{E}_{\tilde{z} \sim \mathscr{N}(0,I)} \Big[$
            $\big( \mathfrak{q}_\theta(s,a,\tilde{z}) - y(r,s') - \bar{\sigma} \langle c, \tilde{z} \rangle \big) \cdot \nabla_{\theta_\sigma} \sigma^L_{\theta_\sigma}(\psi_{\theta_\mu}(s),a,\tilde{z}) \Big] + 2\lambda_\sigma \theta_\sigma$
    Update actor network:
        $\phi \leftarrow \phi + \eta_\pi \cdot \frac{1}{|\bar{\mathscr{B}}|} \sum_{s \in \bar{\mathscr{B}}} \Bigg[ \sum_{a \in \mathscr{A}} \Bigg( \hat{\mu}(s,a) - \rho \cdot \frac{\Sigma_\theta(s) \pi_\phi(a|s)}{\|\Sigma^{1/2}_\theta(s) \pi_\phi(\cdot|s)\|} \Bigg) \nabla_\phi \pi_\phi(a|s)$
            $- \alpha \cdot \nabla_\phi \mathbb{E}_{a \sim \pi_\phi}[\log \pi_\phi(a|s)] \Bigg]$
    Update target network: $\theta' \leftarrow \tau \cdot \theta + (1-\tau) \cdot \theta'$
```

Risk-Sensitive Offline Data Generation

Algorithm 13 : Offline Data Generation via Dynamic Expectile Risk Policies

```
Input : Environment $\mathscr{M}$; risk level $\tau \in (0,1)$; dataset size $N_{\mathscr{D}}$; initial policy
        parameters $\phi$, Q parameters $\theta$, target Q parameters $\theta'$, learning rates $\eta_Q$,
        $\eta_\pi$; exploration rate $\varepsilon$; number of samples $N_s$ for $P(\cdot|s,a)$ approximation
Output : Offline dataset $\mathscr{D}$
Initialize policy parameters $\phi$ and value function parameters $\theta$
for each epoch do
    Reset environment $\mathscr{M}$ and observe state $s$
    while Episode not done do
        Sample transition $(s,a,r,s')$ by executing current policy $\pi_\phi$
        Resample $N_s$ transitions from $(s,a)$ to assemble $\hat{p}_{N_s}(\cdot|s,a)$
        Compute expectile target:
            $$y \leftarrow \sup\Big\{z : \mathbb{E}_{s'\sim\hat{p}_{N_s}(\cdot|s,a)}\Big[\Big|\tau - \mathbb{I}\Big(z < r + \gamma\max_{a'} Q_{\theta'}(s',a')\Big)\Big| \cdot \Big(z - r - \gamma\max_{a'} Q_{\theta'}(s',a')\Big)\Big] \leq 0\Big\}$$
        Update value function:
            $$\theta \leftarrow \theta - \eta_Q \cdot \nabla_\theta \left(Q_\theta(s,a) - y\right)^2$$
        Update policy:
            $$\phi \leftarrow \phi + \eta_\pi \cdot \mathbb{E}_{a\sim\pi_\phi(\cdot|s)}\left[\nabla_\phi \log \pi_\phi(a|s) \cdot Q_\theta(s,a)\right]$$
        Move to next state: $s \leftarrow s'$
    Update target network: $\theta' \leftarrow \tau\theta + (1-\tau)\theta'$
```

Algorithm 13 : (continued)

```
Offline Data Collection with $\varepsilon$-Greedy Exploration:
Initialize empty dataset $\mathscr{D} \leftarrow \emptyset$
while $|\mathscr{D}| < N_{\mathscr{D}}$ do
    Observe state $s$ from environment $\mathscr{M}$
    if RandomUniform(0,1) < $\varepsilon$ then
        Sample action $a \sim \text{Uniform}(\mathscr{A})$
    else
        Sample action $a \sim \pi_\phi(\cdot|s)$
    Execute action $a$ in environment to observe $r$ and $s'$
    Store $(s,a,r,s')$ in buffer $\mathscr{D}$
return Dataset $\mathscr{D}$
```

Training algorithm details

We evaluate all algorithms on a tabular Machine Replacement MDP with $S = 10$ states and $A = 2$ actions. Transition dynamics are defined probabilistically, with increasing expected costs for continued operation and a reset mechanism triggered by replacement actions. Rewards are state- and transition-dependent, with negative values to simulate maintenance costs and catastrophic penalties for failure.

To construct behavior policies, we implement risk-sensitive value iteration using the expectile risk measure at levels $\tau \in \{0.1, 0.5, 0.9\}$. Expectile backups are computed by solving a convex root-finding problem for each state-action pair. Policies are derived via one-hot argmax over the resulting Q-values.

We generate offline trajectories using the expectile-optimal policy π_τ for each τ. At each step, with probability 0.1, a uniformly random action is taken for exploration. We vary the number of transitions $M \in \{100, 1,000, 10,000\}$ and use ten random seeds per setting. Each trajectory entry records (s,a,s',r).

We evaluate three risk-sensitive SAC-N variants using $N = 100$ Q-ensemble members.

Each method includes entropy regularization with coefficient $\alpha = 0.01$ and actor-critic learning rates $\eta_q = \eta_\pi = 0.01$. Target networks are updated using Polyak averaging with $\tau = 0.005$.

We report normalized returns with respect to the optimal and random policies:

$$\text{Normalized Return} = \frac{V_{\text{eval}} - V_{\text{random}}}{V_{\text{optimal}} - V_{\text{random}}},$$

averaged over 1,000 episodes. Returns are discounted with $\gamma = 0.9$. We repeat all experiments across ten seeds and report the mean and standard deviation. All code is implemented in Pytorch and NumPy using vectorized operations. Root-finding in expectile computation uses a bisection method with machine epsilon tolerance.

References

Amini, A., W. Schwarting, A. Soleimany, and D. Rus. 2020. “Deep evidential regression.” *Advances in neural information processing systems* 33:14927–14937.

An, G., S. Moon, J.-H. Kim, and H. O. Song. 2021. “Uncertainty-based offline reinforcement learning with diversified q-ensemble.” *Advances in neural information processing systems* 34:7436–7447.

Ball, P. J., L. Smith, I. Kostrikov, and S. Levine. 2023. “Efficient online reinforcement learning with offline data.” In *International Conference on Machine Learning,* 1577–1594. PMLR.

Ben-Tal, A., D. Den Hertog, and J.-P. Vial. 2015. “Deriving robust counterparts of nonlinear uncertain inequalities.” *Mathematical programming* 149 (1-2): 265–299.

Bertsimas, D., C. McCord, and B. Sturt. 2022. “Dynamic optimization with side information.” *European Journal of Operational Research.*

Blanquero, R., E. Carrizosa, and N. Gómez-Vargas. 2023. “Contextual Uncertainty Sets in Robust Linear Optimization.”

Chen, L., K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch. 2021. “Decision transformer: Reinforcement learning via sequence modeling.” *Advances in neural information processing systems* 34:15084–15097.

Chen, X., Z. Zhou, Z. Wang, C. Wang, Y. Wu, and K. Ross. 2020. “Bail: Best-action imitation learning for batch deep reinforcement learning.” *Advances in Neural Information Processing Systems* 33:18353–18363.

Chenreddy, A. R., N. Bandi, and E. Delage. 2022. “Data-driven conditional robust optimization.” *Advances in Neural Information Processing Systems* 35:9525–9537.

Christodoulou, P. 2019. “Soft actor-critic for discrete action settings.” *arXiv preprint arXiv:1910.07207.*

Esteban-Pérez, A., and J. M. Morales. 2022. “Distributionally robust stochastic programs with side information based on trimmings.” *Mathematical Programming* 195 (1): 1069–1105.

Filos, A., P. Tigas, R. McAllister, Y. Gal, and S. Levine. 2022. “Epistemic Value Estimation for Risk-Averse Offline Reinforcement Learning.” In *Proceedings of the AAAI Conference on Artificial Intelligence,* 36:8073–8081. 8.

Fu, J., A. Kumar, O. Nachum, G. Tucker, and S. Levine. 2020. “D4rl: Datasets for deep data-driven reinforcement learning.” *arXiv preprint arXiv:2004.07219.*

Fujimoto, S., H. Hoof, and D. Meger. 2018. “Addressing function approximation error in actor-critic methods.” In *International conference on machine learning,* 1587–1596. PMLR.

Ghavamzadeh, M., S. Mannor, J. Pineau, and A. Tamar. 2015. “Bayesian Reinforcement Learning: A Survey.” In *Foundations and Trends in Machine Learning,* 8:359–483. 5-6. Now Publishers Inc.

Ghosh, D., A. Ajay, P. Agrawal, and S. Levine. 2022. “Offline rl policies should be trained to be adaptive.” In *International Conference on Machine Learning,* 7513–7530. PMLR.

Goerigk, M., and J. Kurtz. 2020. “Data-Driven Robust Optimization using Unsupervised Deep Learning.” *arXiv preprint arXiv:2011.09769.*

Gulcehre, C., Z. Wang, A. Novikov, T. Paine, S. Gómez, K. Zolna, R. Agarwal, J. S. Merel, D. J. Mankowitz, C. Paduraru, et al. 2020. “Rl unplugged: A suite of benchmarks for offline reinforcement learning.” *Advances in Neural Information Processing Systems* 33:7248–7259.

Haarnoja, T., A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, et al. 2018. “Soft actor-critic algorithms and applications.” *arXiv preprint arXiv:1812.05905.*

Jelley, A., T. McInroe, S. Devlin, and A. Storkey. 2024. “Efficient Offline Reinforcement Learning: The Critic is Critical.” *arXiv preprint arXiv:2406.13376.*

Kendall, A., and Y. Gal. 2017. “What uncertainties do we need in bayesian deep learning for computer vision?” *Advances in neural information processing systems* 30.

Kidambi, R., A. Rajeswaran, P. Netrapalli, and T. Joachims. 2020. “Morel: Model-based offline reinforcement learning.” *Advances in neural information processing systems* 33:21810–21823.

Kostrikov, I., A. Nair, and S. Levine. 2021. “Offline reinforcement learning with implicit q-learning.” *arXiv preprint arXiv:2110.06169.*

Kumar, A., J. Fu, M. Soh, G. Tucker, and S. Levine. 2019. “Stabilizing off-policy q-learning via bootstrapping error reduction.” *Advances in neural information processing systems* 32.

Kumar, A., J. Hong, A. Singh, and S. Levine. 2022. “When should we prefer offline reinforcement learning over behavioral cloning?” *arXiv preprint arXiv:2204.05618.*

Kumar, A., A. Zhou, G. Tucker, and S. Levine. 2020. “Conservative q-learning for offline reinforcement learning.” *Advances in neural information processing systems* 33:1179–1191.

Lakshminarayanan, B., A. Pritzel, and C. Blundell. 2017. “Simple and scalable predictive uncertainty estimation using deep ensembles.” *Advances in neural information processing systems* 30.

Levine, S., A. Kumar, G. Tucker, and J. Fu. 2020. “Offline reinforcement learning: Tutorial, review, and perspectives on open problems.” *arXiv preprint arXiv:2005.01643.*

Marzban, S., E. Delage, and J. Y.-M. Li. 2023. “Deep reinforcement learning for option pricing and hedging under dynamic expectile risk measures.” *Quantitative finance* 23 (10): 1411–1430.

McCord, C. 2019. “Data-driven dynamic optimization with auxiliary covariates.” PhD diss., Massachusetts Institute of Technology.

Nguyen, V. A., F. Zhang, J. Blanchet, E. Delage, and Y. Ye. 2021. *Robustifying conditional portfolio decisions via optimal transport.*

Ohmori, S. 2021. “A Predictive Prescription Using Minimum Volume k-Nearest Neighbor Enclosing Ellipsoid and Robust Optimization.” *Mathematics* 9 (2): 119.

Osband, I., Z. Wen, S. M. Asghari, V. Dwaracherla, M. Ibrahimi, X. Lu, and B. Van Roy. 2023. “Epistemic neural networks.” *Advances in Neural Information Processing Systems* 36:2795–2823.

Panaganti, K., Z. Xu, D. Kalathil, and M. Ghavamzadeh. 2022. “A Risk-Sensitive Perspective on Model-Based Offline Reinforcement Learning.” In *Advances in Neural Information Processing Systems,* 35:12345–12356.

Prudencio, R. F., M. R. Maximo, and E. L. Colombini. 2023. “A survey on offline reinforcement learning: Taxonomy, review, and open problems.” *IEEE Transactions on Neural Networks and Learning Systems.*

Schweighofer, K., M.-c. Dinu, A. Radler, M. Hofmarcher, V. P. Patil, A. Bitto-Nemling, H. Eghbal-zadeh, and S. Hochreiter. 2022. “A dataset perspective on offline reinforcement learning.” In *Conference on Lifelong Learning Agents,* 470–517. PMLR.

Shi, L., and Y. Chi. 2022. “Distributionally Robust Model-Based Offline Reinforcement Learning with Near-Optimal Sample Complexity.” *Journal of Machine Learning Research* 25 (1): 1–46.

Sun, C., L. Liu, and X. Li. 2023. “Predict-then-Calibrate: A New Perspective of Robust Contextual LP.” In *Advances in Neural Information Processing Systems (NeurIPS).* https://proceedings.neurips.cc/paper_files/paper/2023/file/397271e11322fae8ba7f827c50ca8d9b-Paper-Conference.pdf.

Wang, C., and S. Chen. 2020. "A distributionally robust optimization for blood supply network considering disasters." *Transportation Research Part E: Logistics and Transportation Review* 134:101840.

Wang, I., C. Becker, B. Van Parys, and B. Stellato. 2023. "Learning for Robust Optimization." *arXiv preprint arXiv:2305.19225.*

Wen, Y., D. Tran, and J. Ba. 2020. "Batchensemble: an alternative approach to efficient ensemble and lifelong learning." *arXiv preprint arXiv:2002.06715.*

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

Wu, Y., G. Tucker, and O. Nachum. 2019. "Behavior regularized offline reinforcement learning." *arXiv preprint arXiv:1911.11361.*

Yang, Y., X. Ma, C. Li, Z. Zheng, Q. Zhang, G. Huang, J. Yang, and Q. Zhao. 2021. "Believe what you see: Implicit constraint approach for offline multi-agent reinforcement learning." *Advances in Neural Information Processing Systems* 34:10299–10312.

Younis, O. G., R. Perez-Vicente, J. U. Balis, W. Dudley, A. Davey, and J. K. Terry. 2024. *Minari.* V. 0.5.0, September. https://doi.org/10.5281/zenodo.13767625. https://doi.org/10.5281/zenodo.13767625.

Yu, T., A. Kumar, R. Rafailov, A. Rajeswaran, S. Levine, and C. Finn. 2021. "Combo: Conservative offline model-based policy optimization." *Advances in neural information processing systems* 34:28954–28967.

Yu, T., G. Thomas, L. Yu, S. Ermon, J. Y. Zou, S. Levine, C. Finn, and T. Ma. 2020. "Mopo: Model-based offline policy optimization." *Advances in Neural Information Processing Systems* 33:14129–14142.

General Conclusion

This thesis presented Conditional Robust Optimization (CRO), a novel framework for learning context dependent uncertainty sets that enable robust, data driven decisions. Unlike classical robust optimization, which relies on global worst-case uncertainty sets, the CRO paradigm models uncertainty as a function of observable covariates, thus allowing decisions to respond meaningfully to evolving information. Chapter 1 formalized this paradigm, provided formulations with statistical coverage guarantees, and established a foundation for contextual robustness in data driven settings.

Subsequent work has extended CRO in several directions centered around the core insight that robust decisions should also adapt to context. Y. P. Patel, Rayan, and Tewari (2024) apply conformal prediction to construct non-convex uncertainty sets with finite-sample guarantees. Sun, Liu, and Li (2023) propose a predict-then-calibrate framework that decouples prediction from robust optimization in linear programs. J. Yang et al. (2022) introduce causal transport-based sets that preserve conditional dependence structures, while Zhang et al. (2024) leverage high-dimensional support vector clustering to construct localized, feature-dependent sets.

To align uncertainty quantification more tightly with decision objectives, Chapter 2 further developed an end-to-end CRO (E2E-CRO) methodology that jointly learns uncertainty sets and robust decisions via a differentiable surrogate for the robust objective. This contributes to the growing literature on decision-aware uncertainty modeling. For example, Ma, Ning, and Du (2024) propose differentiable DRO layers for mixed-integer programs, Cortes-Gomez et al. (2024) optimize conformal prediction sets for decision utility, and

Jacquillat and Li (2024) study regret-optimal learning in settings with irreversible decisions. I. Wang et al. (2023) propose directly minimizing expected decision loss through a stochastic augmented Lagrangian approach to uncertainty set learning.

In the third chapter, we extended these ideas to the sequential decision-making setting using Epistemic Robust Soft Actor-Critic (ERSAC) model. ERSAC brings conditional robustness to offline reinforcement learning by constructing state dependent uncertainty sets over Q-values that reflect epistemic uncertainty. Rather than relying on large ensembles, ERSAC leverages an Epistemic Neural Network to model rich uncertainty structure while maintaining scalability and differentiability. This enables robust Bellman backups and conservative policy learning in data limited, high stakes environments. ERSAC generalizes the CRO philosophy of adapting robustness to context in dynamic settings where uncertainty evolves along with the trajectory, and decisions must remain safe under limited feedback from the environment.

Looking ahead, several promising directions emerge under the broader theme of contextual uncertainty. A key challenge is to develop generalization guarantees for decision performance when uncertainty sets are learned from finite data. Extending contextual robustness to multi-stage or sequential optimization also remains largely open. Here, uncertainty not only depends on context but can evolve as a function of both past states and decisions, suggesting the need for autoregressive constructions that remain tractable while preserving statistical validity (Malinin and Gales (2020)). Another emerging direction is the integration of fairness constraints into the structure of context dependent sets. This involves ensuring that the coverage and conservativeness of the learned sets do not systematically vary across sensitive groups or features. For example, one could enforce demographic parity in coverage rates, or penalize heterogeneity in set sizes across subpopulations, thereby preventing minority groups from being systematically over or under protected. Incorporating fairness at the level of uncertainty sets requires balancing statistical guarantees with equitable treatment and would expand the applicability of contextual robustness to socially sensitive domains such as healthcare, credit allocation, and personalized decision making (Kuzucu et al. (2023)).

Beyond structured tabular covariates, there is also growing potential in leveraging unstructured and multimodal data such as text, images, or sensor streams to inform uncertainty set construction. Multimodal contexts could allow decision making models to incorporate richer side information (e.g., medical imaging in healthcare, or natural language in recommendation systems) when quantifying epistemic uncertainty. However, integrating high dimensional modalities raises challenges for both statistical validity and computational tractability i.e., coverage guarantees must extend to feature spaces where distances are poorly defined, and scalable learning procedures must be developed to map complex embeddings into tractable uncertainty sets. One promising direction is to combine representation learning with contextual robustness, using pre-trained encoders to extract lower dimensional features while calibrating set construction on the latent space. Successfully incorporating multimodal data could substantially broaden the scope of contextual robustness, enabling its deployment in modern AI systems where decisions increasingly rely on heterogeneous sources of information.

References

An, G., S. Moon, J.-H. Kim, and H. O. Song. 2021. “Uncertainty-based offline reinforcement learning with diversified q-ensemble.” *Advances in neural information processing systems* 34:7436–7447.

Beck, A., and A. Ben-Tal. 2009. “Duality in robust optimization: primal worst equals dual best.” *Operations Research Letters* 37 (1): 1–6.

Ben-Tal, A., B. Do Chung, S. R. Mandala, and T. Yao. 2011. “Robust optimization for emergency logistics planning: Risk mitigation in humanitarian relief supply chains.” *Transportation research part B: methodological* 45 (8): 1177–1189.

Ben-Tal, A., and A. Nemirovski. 2000. “Robust solutions of linear programming problems contaminated with uncertain data.” *Mathematical programming* 88:411–424.

Bennouna, O., J. Zhang, S. Amin, and A. Ozdaglar. 2024. “Addressing misspecification in contextual optimization.” *arXiv preprint arXiv:2409.10479.*

Bertsimas, D., V. Gupta, and N. Kallus. 2018. “Data-driven robust optimization.” *Mathematical Programming* 167:235–292.

Bertsimas, D., and M. Sim. 2003. “Robust discrete optimization and network flows.” *Mathematical programming* 98 (1): 49–71.

Bertsimas, D., and A. Thiele. 2004. “A robust optimization approach to supply chain management.” In *Integer Programming and Combinatorial Optimization: 10th International IPCO Conference, New York, NY, USA, June 7-11, 2004. Proceedings 10,* 86–100. Springer.

Besbes, O., Y. Gur, and A. Zeevi. 2015. “Non-stationary stochastic optimization.” *Operations research* 63 (5): 1227–1244.

Birge, J. R., and F. Louveaux. 1997. *Introduction to stochastic programming.* Springer.

Blanchet, J., J. Li, S. Lin, and X. Zhang. 2024. “Distributionally Robust Optimization and Robust Statistics.” *arXiv preprint arXiv:2401.14655,* https://arxiv.org/abs/2401.14655.

Chen, X., M. Sim, and P. Sun. 2007. “A robust optimization perspective on stochastic programming.” *Operations research* 55 (6): 1058–1071.

Chenreddy, A., and E. Delage. 2024. “End-to-end conditional robust optimization.” *arXiv preprint arXiv:2403.04670.*

Chenreddy, A. R., N. Bandi, and E. Delage. 2022. “Data-driven conditional robust optimization.” *Advances in Neural Information Processing Systems* 35:9525–9537.

Cortes-Gomez, S., C. Patiño, Y. Byun, S. Wu, E. Horvitz, and B. Wilder. 2024. “Decision-focused uncertainty quantification.” *arXiv preprint arXiv:2410.01767.*

Elmachtoub, A. N., H. Lam, H. Zhang, and Y. Zhao. 2023. “Estimate-then-optimize versus integrated-estimation-optimization versus sample average approximation: a stochastic dominance perspective.” *arXiv preprint arXiv:2304.06833.*

Goerigk, M., and J. Kurtz. 2020. “Data-Driven Robust Optimization using Unsupervised Deep Learning.” *arXiv preprint arXiv:2011.09769.*

Jacquillat, A., and M. L. Li. 2024. “Learning to Cover: Online Learning and Optimization with Irreversible Decisions.” *arXiv preprint arXiv:2406.14777.*

Jiang, N., and W. Xie. 2023. “Distributionally Favorable Optimization: A Framework for Data-driven Decision-making with Endogenous Outliers.” *arXiv preprint arXiv:2311.05573,* https://arxiv.org/abs/2311.05573.

Kuzucu, S., J. Cheong, H. Gunes, and S. Kalkan. 2023. “Uncertainty-based fairness measures.” *arXiv preprint arXiv:2312.11299.*

Lam, H. 2019. “Recovering best statistical guarantees via the empirical divergence-based distributionally robust optimization.” *Operations Research* 67 (4): 1090–1105.

Lemke, C., M. Budka, and B. Gabrys. 2015. “Metalearning: a survey of trends and technologies.” *Artificial intelligence review* 44 (1): 117–130.

Li, C., and I. E. Grossmann. 2021. “A review of stochastic programming methods for optimization of process systems under uncertainty.” *Frontiers in Chemical Engineering* 2:622241.

Ma, X., C. Ning, and W. Du. 2024. “Differentiable Distributionally Robust Optimization Layers.” *arXiv preprint arXiv:2406.16571,* https://arxiv.org/abs/2406.16571.

Malinin, A., and M. Gales. 2020. “Uncertainty estimation in autoregressive structured prediction.” *arXiv preprint arXiv:2002.07650.*

Mandi, J., J. Kotary, S. Berden, M. Mulamba, V. Bucarey, T. Guns, and F. Fioretto. 2024. “Decision-focused learning: Foundations, state of the art, benchmark and future opportunities.” *Journal of Artificial Intelligence Research* 80:1623–1701.

Osband, I., Z. Wen, S. M. Asghari, V. Dwaracherla, M. Ibrahimi, X. Lu, and B. Van Roy. 2023. “Epistemic neural networks.” *Advances in Neural Information Processing Systems* 36:2795–2823.

Patel, Y. P., S. Rayan, and A. Tewari. 2024. “Conformal Contextual Robust Optimization.” In *Proceedings of The 3rd Conference on Causal Learning and Reasoning (CLeaR),* 238:593–609. Proceedings of Machine Learning Research. PMLR.

Rosolia, U., X. Zhang, and F. Borrelli. 2018. “Data-driven predictive control for autonomous systems.” *Annual Review of Control, Robotics, and Autonomous Systems* 1 (1): 259–286.

Sadana, U., A. Chenreddy, E. Delage, A. Forel, E. Frejinger, and T. Vidal. 2025. “A survey of contextual optimization methods for decision-making under uncertainty.” *European Journal of Operational Research* 320 (2): 271–289.

Shapiro, A., D. Dentcheva, and A. Ruszczynski. 2021. *Lectures on stochastic programming: modeling and theory.* SIAM.

Smith, J. E., and R. L. Winkler. 2006. “The optimizer’s curse: Skepticism and postdecision surprise in decision analysis.” *Management Science* 52 (3): 311–322.

Sun, C., L. Liu, and X. Li. 2023. “Predict-then-Calibrate: A New Perspective of Robust Contextual LP.” In *Advances in Neural Information Processing Systems (NeurIPS).* https://proceedings.neurips.cc/paper_files/paper/2023/file/397271e11322fae8ba7f827c50ca8d9b-Paper-Conference.pdf.

Suryawanshi, P., and P. Dutta. 2022. “Optimization models for supply chains under risk, uncertainty, and resilience: A state-of-the-art review and future research directions.” *Transportation research part e: logistics and transportation review* 157:102553.

Van Parys, B. P., P. M. Esfahani, and D. Kuhn. 2021. “From data to decisions: Distributionally robust optimization is optimal.” *Management Science* 67 (6): 3387–3402.

Wang, I., C. Becker, B. Van Parys, and B. Stellato. 2023. “Learning for Robust Optimization.” *arXiv preprint arXiv:2305.19225.*

Xidonas, P., R. Steuer, and C. Hassapis. 2020. “Robust portfolio optimization: a categorized bibliographic review.” *Annals of Operations Research* 292 (1): 533–552.

Yang, J., L. Zhang, N. Chen, R. Gao, and M. Hu. 2022. *Decision-making with Side Information: A Causal Transport Robust Approach.* Technical report. Optimization Online. https://optimization-online.org/wp-content/uploads/2022/10/DRO_with_side_info.pdf.

Zhang, L., R. Gao, N. Chen, and M. Hu. 2024. "Data-driven contextual robust optimization based on support vector clustering." *Operations Research Letters* 52:1–8. https://doi.org/10.1016/j.orl.2023.106279.

Zhou, L. 2015. "A survey on contextual multi-armed bandits." *arXiv preprint arXiv:1508.03326.*