University of Liège
Faculty of Applied Sciences
Department of Electrical Engineering & Computer Science

Doctoral dissertation

Artificial Intelligence Techniques for Decision-Making in Market Environments

by Thibaut Théate

June 2023

Academic advisor: Prof. Damien Ernst

Jury members

Professor Quentin Louveaux (President)	University of Liège, Belgium;
Professor Damien Ernst (Advisor)	University of Liège, Belgium;
Professor Guillaume Drion	University of Liège, Belgium;
Professor Olivier Pietquin	University of Lille, France;
Doctor Raphaël Fonteneau	University of Liège, Belgium;
Doctor Alexandre Huynen	Engie, Belgium.
Doctor Antonio Sutera	Haulogy, Belgium;

Acknowledgements

The present PhD thesis is the result of a particularly interesting and rewarding adventure, during which I have been surrounded by remarkable people. This unique journey would not have been as pleasant without them, and I sincerely wish to acknowledge their support...

Family

First and foremost, I would like to express my deepest thanks to my family, without whom this doctoral thesis would not have been possible. Their unconditional love and support have been an invaluable source of strength and motivation, both on good days and in the most challenging times of the demanding PhD journey. In particular, I want to praise my partner Elodie for her true love, her unshakeable faith in me and for her unwavering encouragement. I am also grateful to my parents for their invaluable education making all of this possible, as well as for their consistent and absolute support. With this doctoral thesis, I hope to make them even more proud of the person I have become. Finally, I do not forget to thank my brother Maxime for continuously challenging me, together with my grandparents.

Supervisor

I am deeply grateful to my supervisor and advisor, Professor Damien Ernst, for offering me the opportunity to carry out the present doctoral thesis, as well as for his unwavering support throughout the entire PhD journey. From day one, he has never stopped believing in my abilities and has been a considerable source of encouragement and motivation. His expertise, menthorship and availability have undeniably been key factors to the success of my thesis. Moreover, Damien Ernst is a truly multidisciplinary scientist with a substantial experience, from whom I have learnt about numerous interesting topics and skills.

Colleagues from Montefiore

My doctoral thesis journey would have been much less enjoyable without my numerous great colleagues from Montefiore. First of all, thank you Antoine Dubois for sharing my office as well as for our long discussions to make the world a better place. I also want to honestly thank Sébastien Mathieu, Ioannis Boukas, Adrien Bolland, Antoine Wehenkel, Gilles Louppe and Antonio Sutera for their valuable collaboration in research works. Then, my sincere thanks to Raphael Fonteneau for his kindness and multiple advice during these past years. But also my warm thanks to Victor Dachet, Bardhyl Miftari, Guillaume Derval, Arnaud Delaunoy, Gaspard Lambrechts, Mathias Berger, Laurine Duchesne, Pascal Leroy, Amina Benzerga, Jocelyn Mbenoun, Alireza Bahmanyar, Jonathan Dumas, Nicolas Vecoven, Samy Attaihar, as well as to all others that I forget to mention here.

Haulogy

I would like to acknowledge the Haulogy company for giving me the rewarding opportunity to collaborate on a very interesting project that has greatly enriched my research. Moreover, I thank the team from the *Blacklight Analytics* spin-off, which has since merged with Haulogy. These former colleagues offered me their valuable experience about the life of a PhD student and contributed significantly to the good mood at work: Quentin Gemine, Julien Confetti, Yves Vanaubel, Sébastien Mathieu, Elodie Burtin, Michael Castronovo and Jordan Taelman.

Jury

I also want to express my gratitude to all members of the jury for their interest in my research work, as well as for accepting to allocate their precious time to my thesis. I have no doubts that their insightful comments and constructive criticism will definitely helped me to further improve the quality of the present manuscript.

F.R.S.-FNRS

It is important to me to acknowledge the considerable financial support of the F.R.S.-FNRS. I feel grateful and honoured to have been selected for a 4-year *Research Fellow* grant, and I have done my best to live up to these resources allocated to my PhD thesis.

Others

Finally, I would like to extend my heartfelt thanks to all the people not mentioned hereabove who have supported me in various ways throughout my doctoral thesis journey.

Figure 2: General illustration of the present doctoral thesis entitled *Artificial Intelligence techniques for decision-making in market environments*, as seen by a generative art AI [1]

Abstract

The present time is witnessing growing evidence indicating that Artificial Intelligence (AI) will play a central role in the upcoming major industrial revolution. According to experts, this groundbreaking technology will significantly transform the daily lives of billions of people worldwide. With each passing day, new AI-based solutions are emerging in a wide range of fields, not only automating repetitive tasks but also tackling intricate problems. In parallel, the present era is characterised by the ubiquitous presence of markets, propelled by major trends such as economic liberalisation. This serious evolution has given rise to a plethora of complex decision-making problems, with far-reaching implications for countless individuals across the globe. Therefore, the elaboration of novel, effective solutions to these challenges could yield immense benefits. This situation motivates the scope of the present doctoral thesis, which can be summarised as follows: *The study of complex sequential decision-making problems related to markets, and the development along with analysis of novel algorithmic solutions on the basis of innovative AI techniques.*

The research carried out within the framework of this thesis can be classified into three primary categories: applied research, fundamental research and sustainable research. Firstly, the applied research conducted focuses on the development of novel AI-driven algorithmic solutions aimed at addressing several sequential decision-making problems that arise in the energy and stock markets. Specifically, the Deep Reinforcement Learning (DRL) approach is explored for that purpose. This applied research yields two important contributions. First, innovative solutions to the decision-making problems studied are designed, explained and rigorously evaluated. Second, the analysis of these new algorithmic solutions highlights the potential of the DRL methodology, but also reveals key limitations related to characteristics of market environments. Building upon these observations, a fundamental research is carried out to enhance existing DRL techniques, so that they are more robust to stochastic and poorly observable environments such as markets. In particular, emphasis is placed on the distributional RL approach, which is concerned with the learning of the complete probability distribution of the random return rather than solely its expectation. The main contributions of the fundamental research comprise the introduction of a novel distributional RL algorithm, together with an intuitive methodology for learning risk-sensitive policies. Finally, the author is strongly committed to imbuing this thesis with a more sustainable dimension. To this end, an important decision-making problem in energy markets is rigorously formalised, with the objective to contribute to the improved synchronisation of power consumption and electricity production of intermittent renewable energy sources.

Résumé

À l'heure actuelle, de plus en plus d'éléments indiquent que l'Intelligence Artificielle (IA) jouera un rôle majeur dans l'importante révolution industrielle à venir. Selon les experts, cette technologie novatrice transformera considérablement la vie quotidienne de milliards d'êtres humains. Chaque jour, de nouvelles solutions basées sur l'IA émergent dans un large éventail de domaines, permettant non seulement d'automatiser des tâches répétitives mais aussi de résoudre des problèmes complexes. Parallèlement, notre époque est caractérisée par l'omniprésence des marchés, sous l'impulsion de tendances majeures telles que la libéralisation économique. Cette évolution a donné naissance à une multitude de problèmes complexes de prise de décision, avec d'importantes implications pour d'innombrables individus à travers le monde. Par conséquent, l'élaboration de solutions innovantes et efficaces à ces défis pourrait générer d'immenses bienfaits. Ce constat motive la portée de la présente thèse de doctorat, qui peut être résumée comme suit : *Le développement et l'analyse de solutions algorithmiques innovantes tirant profit de techniques d'IA avancées pour résoudre des problèmes complexes de prise de décision séquentielle relatifs aux marchés.*

Les recherches menées dans le cadre de cette thèse s'inscrivent dans ces trois catégories principales : les recherches appliquée, fondamentale et à caractère durable. Tout d'abord, la recherche appliquée se concentre sur le développement de nouvelles solutions algorithmiques pilotées par l'IA pour résoudre des problèmes de prise de décision séquentielle qui se posent dans les marchés de l'énergie et de la bourse. En particulier, l'approche *Deep Reinforcement Learning* (DRL) est explorée. Cette recherche appliquée produit deux contributions notables. Premièrement, des solutions innovantes aux problèmes étudiés sont conçues, commentées et rigoureusement évaluées. Deuxièmement, l'analyse de ces nouvelles solutions algorithmiques met en évidence le potentiel de l'approche DRL, mais révèle également ses principales limites liées aux caractéristiques des environnements de marché. Sur la base de ces observations, une recherche fondamentale est menée pour améliorer les techniques DRL existantes, afin qu'elles soient plus robustes face à des environnements stochastiques et peu observables tels que les marchés. Plus précisément, l'accent est mis sur l'approche *distributional RL*, qui s'intéresse à l'apprentissage de la distribution de probabilité complète des récompenses plutôt qu'à sa seule espérance. Les principales contributions de cette recherche fondamentale comprennent un nouvel algorithme de ce type, ainsi qu'une méthodologie intuitive pour l'apprentissage de politiques décisionnelles sensibles au risque. Enfin, il est important pour l'auteur de donner à sa thèse une dimension plus durable. A cette fin, un défi clé dans les marchés de l'énergie est formalisé, avec l'objectif de contribuer à une meilleure synchronisation de la consommation d'énergie et de la production d'électricité à partir de sources d'énergie renouvelables.

Contents

1 Introduction **17**
- 1.1 Preliminary context . . . 18
- 1.2 Outline of the thesis . . . 22
- 1.3 Scientific publications . . . 26

2 An Artificial Intelligence Solution for Electricity Procurement in Forward Markets **29**
- 2.1 Introduction . . . 32
- 2.2 Literature review . . . 33
- 2.3 Problem formalisation . . . 34
  - 2.3.1 Input of a procurement policy . . . 36
  - 2.3.2 Output of a procurement policy . . . 36
  - 2.3.3 Objective criterion . . . 37
- 2.4 Algorithmic solution . . . 38
  - 2.4.1 Basic forecaster . . . 42
  - 2.4.2 DL forecaster . . . 42
- 2.5 Performance assessment methodology . . . 44
- 2.6 Results . . . 46
- 2.7 Discussion . . . 50
- 2.8 Future work . . . 51
- 2.9 Conclusions . . . 54

3 An Application of Deep Reinforcement Learning to Algorithmic Trading **58**
- 3.1 Introduction . . . 60
- 3.2 Literature review . . . 61
- 3.3 Algorithmic trading problem formalisation . . . 62
  - 3.3.1 Algorithmic trading . . . 62
  - 3.3.2 Timeline discretisation . . . 63
  - 3.3.3 Trading strategy . . . 63
  - 3.3.4 Reinforcement learning problem formalisation . . . 64
- 3.4 Deep reinforcement learning algorithm . . . 73
  - 3.4.1 Deep Q-Network algorithm . . . 73
  - 3.4.2 Generation of artificial trajectories . . . 73
  - 3.4.3 TDQN algorithm . . . 75

3.5 Performance assessment methodology . . . 78
3.5.1 Benchmark stock markets . . . 78
3.5.2 Benchmark trading strategies . . . 79
3.5.3 Quantitative performance assessment . . . 80
3.6 Results and discussion . . . 81
3.6.1 Positive results - Apple stock . . . 82
3.6.2 Mitigated results - Tesla stock . . . 84
3.6.3 Global results - Testbench . . . 86
3.6.4 Discussion about the discount factor . . . 88
3.6.5 Discussion about the trading costs . . . 88
3.6.6 Challenges identified . . . 90
3.7 Conclusion . . . 91

4 Distributional Reinforcement Learning with Unconstrained Monotonic Neural Networks 94
4.1 Introduction . . . 97
4.2 Literature review . . . 98
4.3 Distributional Reinforcement Learning . . . 100
4.4 Unconstrained monotonic deep Q-network . . . 102
4.4.1 Learning different representations of a probability distribution . . . 102
4.4.2 Unconstrained monotonic neural network . . . 106
4.4.3 Unconstrained monotonic deep Q-network algorithm . . . 108
4.5 Results . . . 114
4.5.1 Benchmark environments . . . 114
4.5.2 Experiments reproducibility . . . 118
4.5.3 Results discussion . . . 122
4.6 Conclusions . . . 129

5 Risk-Sensitive Policy with Distributional Reinforcement Learning 133
5.1 Introduction . . . 135
5.2 Literature review . . . 136
5.3 Theoretical background . . . 137
5.3.1 Markov decision process . . . 137
5.3.2 Distributional reinforcement learning . . . 137
5.4 Methodology . . . 139
5.4.1 Objective criterion for risk-sensitive RL . . . 139
5.4.2 Practical modelling of the risk . . . 139
5.4.3 Risk-based utility function . . . 140
5.4.4 Risk-sensitive distributional RL algorithm . . . 141
5.5 Performance assessment methodology . . . 143
5.5.1 Benchmark environments . . . 143
5.5.2 Risk-sensitive distributional RL algorithm analysed . . . 145
5.6 Results . . . 147
5.6.1 Decision-making policy performance . . . 147
5.6.2 Probability distribution visualisation . . . 151

5.7 Conclusion 152

6 Matching of Everyday Power Supply and Demand with Dynamic Pricing: Problem Formalisation and Conceptual Analysis 155
6.1 Introduction 158
6.2 Literature review 159
6.3 Problem formalisation 160
6.3.1 Contextualisation 160
6.3.2 Decision-making process overview 161
6.3.3 Timeline discretisation 162
6.3.4 Dynamic pricing policy 162
6.3.5 Input of a dynamic pricing policy 163
6.3.6 Output of a dynamic pricing policy 164
6.3.7 Objective criterion 164
6.4 Algorithmic components discussion 167
6.4.1 Power production forecasting 168
6.4.2 Power consumption forecasting 169
6.4.3 Market price forecasting 170
6.4.4 Demand response modelling 171
6.4.5 Uncertainty discussion 172
6.5 Performance assessment methodology 172
6.6 Conclusion 174

7 Conclusion 178
7.1 Key contributions 179
7.2 Future work 180
7.3 Author's closing words 182

The only true wisdom is in knowing you know nothing.

— Socrates

Chapter 1

Introduction

Figure 1.1: Illustration of the introduction chapter of the thesis, created by a generative art AI [1].

1.1 Preliminary context

Nowadays, *Artificial Intelligence* (AI) is on everyone's lips. With the advent of Big Data and the rise of computational power, AI has seen remarkable progress in various key areas including natural language processing, computer vision or decision-making. From healthcare to finance, AI has truly successful applications spanning across numerous fields, not only automating mundane tasks but also solving complex problems. According to many experts, AI is a true game-changer which is expected to shape the next industrial revolution. The daily lives of billions of people will undoubtedly be profoundly transformed. Naturally, such an influential revolution comes with its major concerns and challenges. Overall, AI has the potential to bring significant benefits to society, by improving efficiency, productivity, and quality of life. However, it is capital to approach the development and deployment of AI in a responsible and ethical manner, to ensure that the benefits are shared equitably and that the potential negative impacts are properly addressed.

More formally, AI refers to the ability of computer systems to perform tasks that typically require human intelligence, such as learning, reasoning, and problem-solving. A key driver of innovation in AI is the *Machine Learning* (ML) subfield. This family of techniques involves the development of algorithms and statistical models that enable machines to automatically learn and improve from experience, without being explicitly programmed. In other words, ML algorithms are designed to autonomously identify patterns and insights within large datasets, in order to make predictions or take actions in similar situations. Therefore, the performance of a ML model is not only dependent on the learning algorithm, but also on the quality of the data available for training. As illustrated in Figure 1.2, ML techniques can be subdivided into three main categories: supervised learning, unsupervised learning and reinforcement learning.

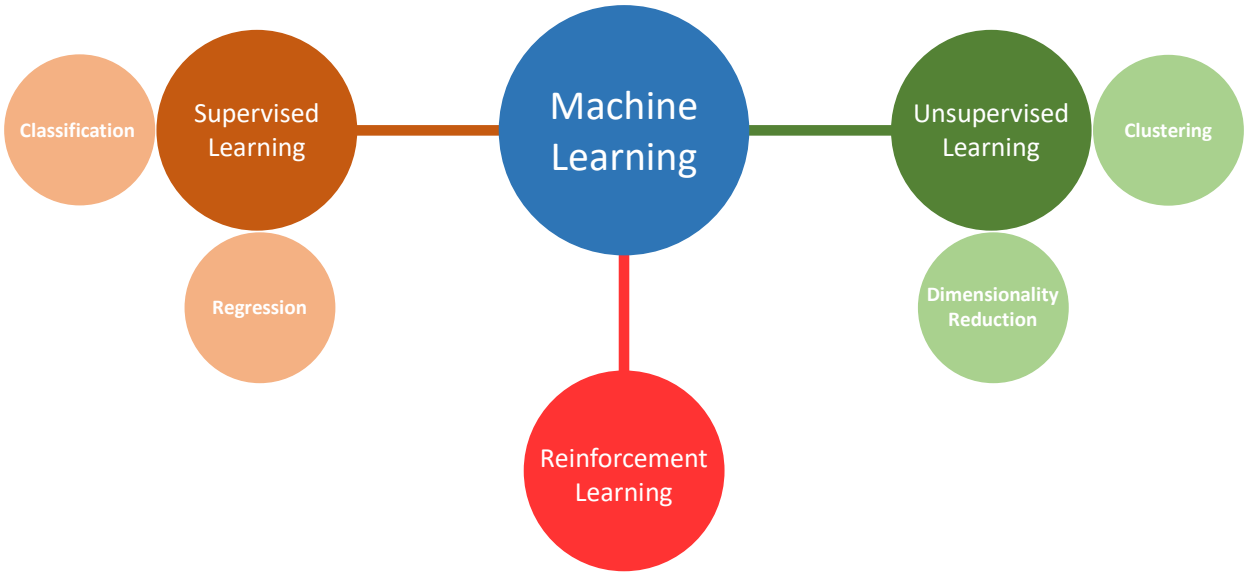


Figure 1.2: Non-exhaustive taxonomy of ML techniques.

In fact, ML techniques are essentially inspired by the learning behaviour of humans and other animals. For instance, supervised learning algorithms mimic the way humans learn from examples. When an unfamiliar object is presented to a human being, he or she learns to recognise this object by observing its features and by associating the findings with a label or name. Unsupervised learning techniques are also inspired by the way humans learn. When confronted with a new environment or data set, a human being tries to identify patterns and structures in the data, without receiving explicit labels or feedback. The reinforcement learning approach takes its inspiration from the way animals extensively learn through trial-and-error. When an animal is presented with a new task, it tries different actions and learns from the feedback it receives, which can either be a positive reward for appropriate behaviour or a negative punishment for making a mistake. Biologically, rewards are generally represented by dopamine, a neurotransmitter playing a key role in the brain's reward system.

Firstly, *Supervised Learning* (SL) is a methodology concerned with the training of a model based on a labelled dataset that contains both input data and corresponding output labels. More precisely, the objective of a SL algorithm is to learn a mapping function from the input data to the output labels, so that it can accurately predict the output labels for new, unseen input data. To achieve that goal, the SL algorithm adjusts the internal parameters of the model during training based on the difference between the predicted outputs and the actual labelled outputs in the training data, with the aim of minimising the prediction error. The SL approach can be adopted for both classification and regression tasks, where the goal is to predict a categorical label or a continuous value, respectively. Examples of SL algorithms include decision trees, support vector machines, and neural networks (NNs).

Secondly, *Unsupervised Learning* (UL) focuses on the training of an algorithm on the basis of an unlabelled dataset, without any output labels available. The goal of this approach is to identify patterns and structures within the input data, such as clusters or groups of similar data points, or underlying features that are not directly observable. Since there is no target variable to predict, the performance of UL algorithms is evaluated by measuring the quality of the discovered patterns or how well the structure of the input data is captured. There exist two main types of UL algorithms: clustering and dimensionality reduction. While clustering techniques group similar data points together into clusters, dimensionality reduction algorithms transform high-dimensional data into a lower-dimensional space. Examples of UL algorithms include k-means clustering, principal component analysis, and autoencoders.

Thirdly, *Reinforcement Learning* (RL) is an approach concerned with the training of an agent by interacting with its environment through trial-and-error to achieve a specific goal. More precisely, the RL agent receives feedback in the form of rewards (or punishments) from the environment as a consequence of the actions performed. The objective is to learn the optimal decision-making policy maximising the cumulative reward over time. The RL methodology is generally considered to solve sequential decision-making problems, in which the agent has to take a series of actions in order to achieve a long-term goal. In this case, the sequential decision-making problem is commonly framed as a *Markov Decision Process* (MDP). Examples of RL algorithms include policy gradient, Q-learning, and actor-critic.

Besides that, *Deep Learning* (DL) is an important subset of ML involving the training of *Deep Neural Networks* (DNNs), from both supervised and unsupervised data. In this case, the mathematical model is an artificial neural network with multiple layers, which is inspired from the structure and function of the human brain. Each layer of the neural network consists of a set of neurons that are connected to the neurons from both the previous and subsequent layers. The input layer receives the raw data, and each subsequent layer processes the data in increasingly abstract and complex ways. The output layer produces the final result of the model, such as a classification or regression prediction. A substantial number of layers enables the learning of complex hierarchical representations of data. The *deep* nomination comes from this feature. Nowadays, the DL techniques achieve state-of-the-art performance on numerous ML tasks such as image/speech recognition and natural language processing.

To finish with the main ML subfields, *Deep Reinforcement Learning* (DRL) combines both DL and RL in order to improve the scalability of the RL approach. Indeed, while the classical RL algorithms are generally limited to low-dimensional decision-making problems, the DRL methodology allows to properly handle high-dimensional state and action spaces. More precisely, DNNs are used to approximate the value function or policy of an agent. Nevertheless, effectively training DNNs in a RL setting is particularly challenging because of the instability of the learning process and the high computational requirements. Still, the DRL approach is the state of the art for a range of challenging tasks, including playing complex board and video games or controlling robotic arms.

On the basis of the ML techniques previously introduced, multiple impressive successes have been achieved by AI over the past decades. In March 2016, *AlphaGo* becomes the first AI program to defeat the world champion Go player, demonstrating the potential for AI to master complex strategy games that were previously thought to be exclusive to human intelligence. Based on an advanced DRL algorithm, this breakthrough is recognised by the research community as an important milestone in AI. Another great example of AI success is *ChatGPT*, released in 2022. Nowadays, it is becoming difficult to find someone who has never heard of or tried ChatGPT, so great is its success. This AI language model, which is built on top of the GPT-3.5 architecture, is very proficient in providing human-like responses to a wide range of questions and topics. In practice, the users mostly view ChatGPT as a very powerful tool for a variety of purposes, from providing quick answers to factual questions, to engaging in more in-depth conversations on a wide range of topics. Over the months, this AI has progressively become a particularly valuable resource for both individuals and businesses alike. In addition to these influential achievements, there has been a recent surge in the development of AI artists. These algorithmic solutions are able to generate highly realistic and detailed images of objects and scenes that don't exist in the real world. Notable examples include *DALL-E*, *MidJourney* or *Stable Diffusion*. In fact, the success of AI in creating visually stunning images that are almost indistinguishable from real photographs is a testament to the power and potential of AI in the field of visual arts, but also in the creative process as a whole. Lastly, it is worth noting that AI has achieved numerous other successes in image/speech recognition, natural language processing, autonomous vehicles, medical diagnosis, fraud detection, among others.

In the contemporary era, propelled by some major trends such as economic liberalisation, markets have become ubiquitous. Whether this direction, as known today, benefits the whole population is an interesting research question, which is not sufficiently debated according to the author. However, this inquiry is beyond the scope of this doctoral thesis, which focuses on the present situation. Nowadays, the markets exhibit distinct mechanisms for the exchange of a diverse range of goods and services, and have a major impact on a tremendous number of people across the world. The proliferation of these different market types has given rise to a host of challenging sequential decision-making problems. Finding innovative and effective solutions to these new challenges would significantly impact people's lives, both socially and economically. This situation represents a great opportunity to explore advanced AI-based algorithmic solutions through research and development. Nevertheless, it is worth noting that markets environments are particularly challenging to deal with, because of their limited observability together with their stochastic and adversarial nature.

Formally, the energy is defined as a physical quantity describing the ability of a system to do work. In practice, the energy may be viewed as the catalyst for transformation in the world, forming the foundation of all things. Therefore, energy is undeniably capital to human society and warrants special attention. In the 1990s, the European Union started the liberalisation of the energy sector, creating a more market-driven system. Once again, the present doctoral thesis does not provide any commentary on this choice, which has its benefits and drawbacks. Nowadays, there exist several different energy (electricity) markets, each with their own unique characteristics and purposes. The forward/future markets allow to trade energy at a future date and at a predetermined price (bilateral contracts), up to several years before the actual delivery. The day-ahead market is operated once a day through a single blind auction, for the exchange of energy delivered on the following day. The continuous intraday market enables market participants to adjust their positions close to real-time, typically on the same day as delivery. The regulation market is operated by the transmission system operator and allows to trade regulating power, in order to ensure the stability of the power grid. Lastly, the imbalance settlement involves payments for potentially contributing to the power grid imbalance, concerning both producers and consumers. As a summary, Figure 1.3 draws a timeline of the primary energy markets in Europe.

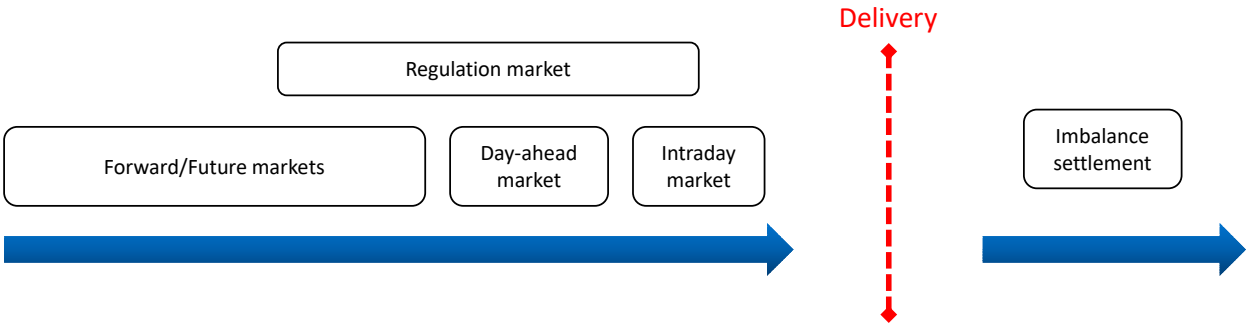


Figure 1.3: Timeline of the primary energy markets in Europe.

1.2 Outline of the thesis

At first, Chapter 1 serves as an introductory section motivating the research conducted. On the basis of the situation previously described, it appears that there is a real opportunity to take advantage of innovative AI techniques in the context of market environments. This is the scope of the present doctoral thesis, whose contribution can be summarised as follows. This research work studies complex sequential decision-making problems related to markets, for which novel algorithmic solutions based on new AI techniques are developed and analysed. Nevertheless, such a project is particularly intricate, since market-related decision-making problems generally have a sequential nature and are highly stochastic, with an environment partially observable and potentially adversarial. Still, the challenge is worthwhile. Given the particularly large number of individuals affected by markets on a daily basis, the social, economical and environmental benefits resulting from this research may be significant.

In the scope of this thesis, both applied and fundamental research has been carried out. More precisely, the research activities conducted may be subdivided into three primary parts. Firstly, some novel AI-based algorithmic solutions are developed for several decision-making problems specific to both energy and stock markets. In particular, the promising DRL methodology is investigated for that purpose. The analysis of these new solutions reveals the major potential of the proposed approach, but also highlights key limitations related to characteristics of market environments. Secondly, on the basis of these observations, more fundamental research is conducted, with the objective to improve existing DRL algorithms so that they are better suited to market environments. To be more specific, contributions on the distributional RL approach are presented. This technique not only makes it possible to better manage stochastic environments, but also to learn risk-sensitive decision-making policies, which are essential in markets. Lastly, the author is particularly keen to give this thesis a sustainable dimension. To achieve that, the formalisation of a key decision-making problem in energy markets is performed, which contributes to the synchronisation of power consumption and electricity production of intermittent renewable energy sources.

To begin with, Chapter 2 studies a challenging decision-making problem related to the forward energy markets. In order to hedge against short-term price volatility, both retailers and large consumers of electricity generally purchase an important share of their estimated energy needs years in advance via the forward markets. This long-term procurement task requires identifying an appropriate timing for power purchases, so that energy costs are minimised while accounting for the projected consumption. In order to address this sequential decision-making problem, the thesis presents a novel algorithmic solution generating daily recommendations to make a purchase now or hold off, on the basis of the history of forward prices. The promising DRL approach was initially explored for that purpose, but without conclusive success. The main reasons for this setback have been identified and are elaborated on further in the manuscript. As an alternative, the research work introduces a method that relies on DL forecasting techniques along with an intuitive mathematical indicator quantifying the deviation from a perfectly uniform procurement policy of reference. The core idea is to distribute the purchase operations over the procurement horizon to spread the trading risk, with nominal anticipation or delay depending on the market direction.

Following that, Chapter 3 focuses on the most popular decision-making problem in the stock markets: algorithmic trading. To be more specific, the objective is to identify the optimal trading position, either long or short, for a particular stock at any moment in time during a trading activity. For a trading strategy to be effective, it has to not only consistently generate profit, but also mitigate the associated risk. The thesis presents a novel algorithmic solution taking advantage of innovative DRL techniques to solve this challenging sequential decision-making problem. Firstly, the algorithmic trading problem is discussed, formalised and framed as a RL problem. In short, the goal is to learn by reinforcement a trading policy so as to maximise the Sharpe ratio performance indicator. Secondly, a new DRL algorithm is designed, which is inspired from the popular DQN algorithm and customised to best suit the specific algorithmic trading problem studied. The resulting decision-making policies are trained solely on the basis of artificial trajectories derived from a restricted collection of historical stock market data. Promising results are reported for the proposed solution, which demonstrates key benefits compared to more classical methods, such as an appreciable versatility and a remarkable robustness to varying trading costs. In addition, the realistic experiments highlight the primary limitations of the DRL for dealing with market environments.

Besides that, an instructive collaboration has been undertaken in the scope of this thesis. However, since this collaboration is the main contribution of another dissertation, it will not be comprehensively discussed within this manuscript. To obtain further details regarding that research, in addition to the following summary, please consult the original article referenced in Section 1.3. The research work presents a novel modelling framework for the strategic participation of energy storage in the continuous intraday electricity market. In short, the objective is the maximisation of the profits earned by the storage device operator, while considering the physical constraints. Firstly, the sequential decision-making problem studied is thoroughly discussed and modelled as an MDP. Secondly, a DRL approach is adopted for solving this problem with an asynchronous version of the fitted Q iteration algorithm. Not delving into specifics, interesting results have been produced. Despite the overall success, the DRL approach appears to face more challenges in such a complex market environment.

On the basis of the observations from the three research works previously described, the primary limitations of the DRL approach in the context of market environments could be identified. Without going into detail, the key challenges can be summarised as the following. Firstly, from the perspective of a RL agent, markets are generally stochastic environments that are poorly observable. Such characteristics require a more robust decision-making in the face of uncertainty. Secondly, the dynamics of market environments are continuously changing, which may significantly impact the performance of AI-based solutions requiring the training dataset to be representative of the test dataset. Thirdly, a successful decision-making in market environments is generally requested to not only maximise the expected performance, as typically seeked in RL, but also to properly mitigate the risk. Lastly, the poor interpretability of DRL black-box models hurts their reliability, especially in markets. Acknowledging these obstacles, the second part of the thesis consists of a more fundamental research about distributional RL, which contributes to improve the situation.

Subsequently, Chapter 4 presents valuable contributions to the distributional RL field. At its core, the distributional RL methodology suggests representing the entire probability distribution of the random return instead of solely modelling its expectation. This approach presents major advantages that are key to market environments, at the expense of increased complexity and computational costs. Among others, it allows for a more accurate modelling of the environment and makes risk-sensitive control and exploration policies possible. The primary concept driving the research conducted is the consideration of an innovative DL architecture, which has been demonstrated to be a universal approximator of continuous monotonic functions. This feature is particularly convenient for effectively modelling different representations of a probability distribution, which is essential in distributional RL. Taking advantage of this architecture, a novel distributional RL method is introduced, supporting the learning of three, valid and continuous representations of the random return distribution. In light of this new algorithm, an empirical evaluation is carried out to fairly compare three probability quasi-metrics typically used in distributional RL for three different distribution representations, and highlight their respective strengths and weaknesses.

Then, Chapter 5 presents an intuitive approach to learn risk-sensitive decision-making policies on the basis of the distributional RL methodology. In most market environments, properly mitigating the risk associated with the trading activities is capital. Nevertheless, traditional RL techniques generally focus on the learning of policies solely maximising the expected outcome. In order to achieve risk-sensitive RL, the research work introduces a new risk-based utility function, taking into consideration both the expected return and the risk, to be maximised. Replacing the popular state-action value function, this utility is to be derived from the complete probability distribution of the random return learnt by any distributional RL algorithm. Ultimately, the objective of this novel risk-sensitive RL approach is to provide an intuitive, straightforward, and accessible solution that prioritises the interpretability of the decision-making policies learnt.

Afterwards, Chapter 6 consists of the last important part of the doctoral thesis, including a contribution to greater sustainability in the energy sector, while being related to markets. The author's motivation for conducting this supplementary research is rooted in his desire to contribute to the energy transition and combat the devastating effects of climate change and biodiversity loss, which are undeniable challenges facing humanity in the 21st century. The research work presents a novel method for improving the synchronisation of power supply and demand, by leveraging the flexibility provided by the demand side. It is necessary because of the increasing share of intermittent renewable energy sources in the electricity mix. The proposed solution, denominated dynamic pricing, consists in providing the consumer with a price signal that is continuously evolving over time, in order to influence its consumption. It involves a challenging decision-making problem to be faced by power producers or retailers: choosing a price signal that maximises the synchronisation of power supply and demand, under the constraints of balancing the profitability of the producer with the benefits to the end consumer. This thesis presents a detailed formalisation of this particular decision-making problem, together with a conceptual analysis of the necessary algorithmic components.

Finally, Chapter 7 concludes the doctoral thesis. The main contributions of the overall research work are summarised and discussed. Moreover, multiple avenues are suggested as future work, in order to continue the research initiated by this thesis.

While there is a coherent connection between the different chapters of the doctoral thesis, each chapter is designed to be comprehensible independently without requiring the reader to go through the entire manuscript. For this reason, some redundancies may exist across the chapters, intentionally preserved to ensure clarity.

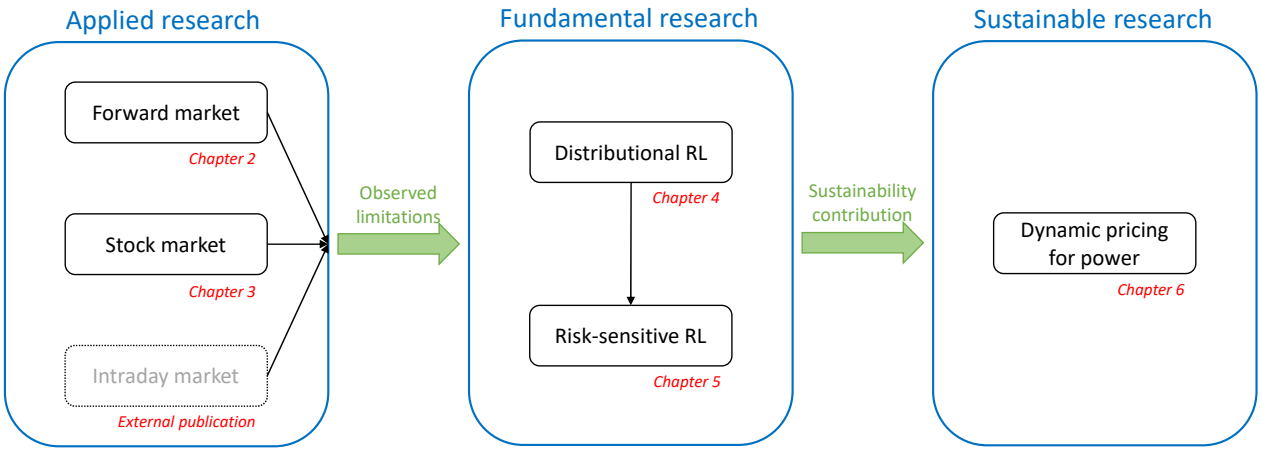


Figure 1.4: Illustration of the outline of the doctoral thesis.

1.3 Scientific publications

In the scope of this doctoral thesis, numerous research activities have been carried out, resulting in several publications in renowned scientific journals. Thoroughly discussed within the manuscript are the following publications of which I am the main author:

- Thibaut Théate, Sébastien Mathieu, and Damien Ernst. An Artificial Intelligence Solution for Electricity Procurement in Forward Markets. *Energies*, 13(23), 2020.
  → **Chapter 2 and Reference [2].**

- Thibaut Théate and Damien Ernst. An Application of Deep Reinforcement Learning to Algorithmic Trading. *Expert Systems with Applications*, 173:114632, 2021.
  → **Chapter 3 and Reference [3].**

- Thibaut Théate, Antoine Wehenkel, Adrien Bolland, Gilles Louppe, and Damien Ernst. Distributional Reinforcement Learning with Unconstrained Monotonic Neural Networks. *Neurocomputing*, 534:199–219, 2023.
  → **Chapter 4 and Reference [4].**

- Thibaut Théate and Damien Ernst. Risk-Sensitive Policy with Distributional Reinforcement Learning. *Algorithms*, 16(7):325, 2023.
  → **Chapter 5 and Reference [5].**

- Thibaut Théate, Antonio Sutera, and Damien Ernst. Matching of Everyday Power Supply and Demand with Dynamic Pricing: Problem Formalisation and Conceptual Analysis. *Energy Reports*, 9:2453–2462, 2023.
  → **Chapter 6 and Reference [6].**

Additionally, a fruitful collaboration took part during the course of this PhD thesis, that has led to the following publication of which I am a co-author. Despite being related to this particular dissertation as well, it is not discussed in detail within the manuscript:

- Ioannis Boukas, Damien Ernst, Thibaut Théate, Adrien Bolland, Alexandre Huynen, Martin Buchwald, Christelle Wynants, and Bertrand Cornélusse. A Deep Reinforcement Learning Framework for Continuous Intraday Market Bidding. *Machine Learning*, 110(9):2335–2387, 2021.
  → **Reference [7].**

If I have seen further it is by standing on the shoulders of giants.

— Isaac Newton

Chapter 2

An Artificial Intelligence Solution for Electricity Procurement in Forward Markets

Figure 2.1: Illustration of Chapter 2 entitled *An Artificial Intelligence Solution for Electricity Procurement in Forward Markets*, created by a generative art AI [1].

Chapter overview

Both retailers and large consumers of electricity generally purchase an important percentage of their estimated electricity needs years ahead in the forward markets, to hedge against short-term price volatility. This long-term electricity procurement task involves determining when to buy power over a certain trading horizon so that the resulting energy costs are minimised, while covering the predicted consumption. In this research work, the experimentation is based on a yearly base load product from the Belgian forward market, named *Calendar* (CAL), which is tradable up to three years ahead of the delivery period. To solve this sequential decision-making problem, this thesis chapter presents a novel algorithm providing recommendations to either buy electricity now or wait for a future opportunity, on the basis of the history of CAL prices. More precisely, the algorithmic solution proposed relies on deep learning forecasting techniques together with an indicator quantifying the deviation from a perfectly uniform procurement policy of reference. On average, this approach surpasses the benchmark procurement policies taken into consideration. Moreover, a reduction in costs of 1.7% with respect to the average electricity price over the trading horizon is achieved. Finally, among other benefits, the generality of the algorithmic solution presented makes it particularly convenient for solving procurement decision-making problems related to other commodities.

This thesis chapter is primarily based on the following scientific publication [2]:

Thibaut Théate, Sébastien Mathieu, and Damien Ernst. An Artificial Intelligence Solution for Electricity Procurement in Forward Markets. *Energies*, 13(23), 2020.

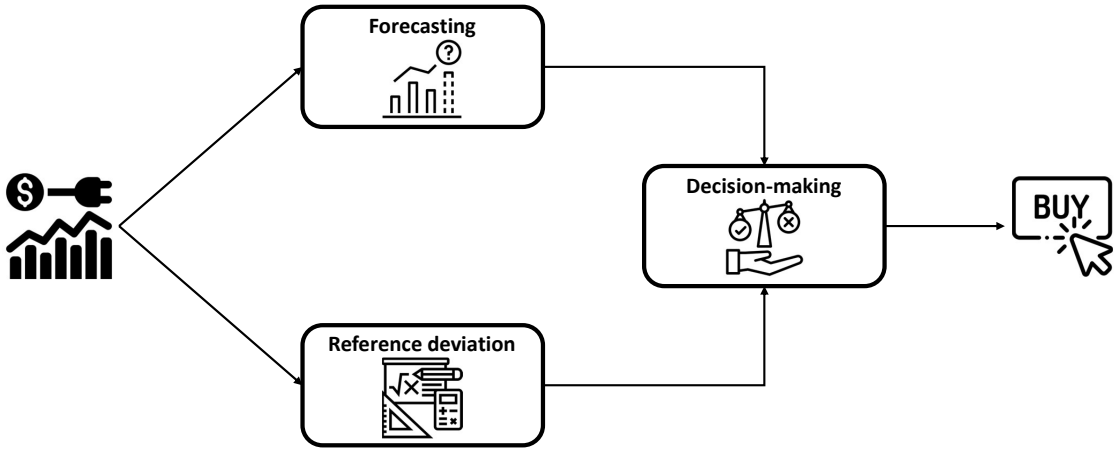


Figure 2.2: General illustration of the methodology presented in this thesis chapter entitled *An Artificial Intelligence Solution for Electricity Procurement in Forward Markets.*

2.1 Introduction

An energy retailer faces a challenging sequential decision-making problem on a daily basis. After having accurately estimated the consumption of its clients, this actor has to purchase the appropriate quantity of electricity in the various energy markets. In order to mitigate the risk associated with the trading activity, the electricity retailers generally buy an important share of their power consumption years ahead in the forward markets, limiting their exposure to short-term markets (hedging). This challenging task also applies to large power consumers that have flexible bilateral contracts with their energy retailer. Typically, they have to decide when to purchase blocks of electricity at a price generally indexed on the forward price. Each block corresponds to a certain percentage of their total electricity consumption, this quantity being formerly predicted by the retailer. Eventually, the potential discrepancy between the power purchased and the predicted consumption is covered by the retailer at the end of the procurement horizon. To summarise, the long-term electricity procurement problem consists in determining when to purchase power in the forward markets, so that the predicted energy needs are secured and the energy costs are minimised. This key decision-making problem is particularly challenging because of its sequential and highly stochastic nature, coupled with a poorly observable market environment.

In practice, the long-term power procurement task is generally entrusted to experienced consultants, on the basis of customised rules together with their expectations regarding the future direction of energy markets. This research work introduces an alternative approach: an algorithmic solution providing recommendations to either buy electricity now or to wait for a future opportunity, based on the history of forward prices. To the authors' opinion, this novel solution may not only interest these consultants, but also retailers who are willing to deploy more advanced procurement techniques and large consumers choosing not to rely on consultants for buying their electricity. Therefore, the core objective of this research work can be summarised as the following: the design of a new relevant algorithmic solution meeting the criteria of the industry, such as the robustness and interpretability of the results. Even though the proposed approach is tailor-made for the purchase of electricity, it can be easily adapted to other commodities as well.

Without going into detail, the algorithm designed is based on the idea that the purchase decisions should ideally be split over the procurement horizon to spread the trading risk, with a nominal anticipation or delay depending on the expected market direction. The proposed solution relies on a forecasting mechanism to predict the future dominant market trend and on an indicator quantifying the deviation from a reference perfectly uniform procurement policy, in order to trigger purchase decisions. In addition to conventional methodologies, *deep learning* (DL) techniques are taken into consideration for the forecasting task, since *deep neural networks* (DNNs) have been shown to effectively manage temporal dependence and structures like trends. However, despite forecasting techniques being a key component of the algorithmic solution designed, this thesis chapter does not focus on the improvement of the forecasts related to forward prices. Instead, the main contribution of this research work is related to the full decision-making process. In other words, how to make relevant trading decisions on the basis of imperfect information, including inaccurate forecasts?

2.2 Literature review

On the one hand, the scientific literature covers multiple techniques for power producers willing to sell their energy in the forward markets. On the other hand, the side of the power retailers/consumers lacks proper scientific coverage, with only a few articles available at the time of writing. In brief, the algorithmic solutions promoted are mostly based on stochastic programming and optimisation techniques.

Research paper [8] introduces a solution to the electricity procurement problem faced by a large consumer whose supply sources include bilateral contracts, self-production and the day-ahead market. More precisely, a stochastic programming approach is adopted, with risk aversion being modelled using the *Conditional Value at Risk* (CVaR) methodology. The proposed solution is assessed through a realistic case study that highlights the trade-off between cost minimisation and risk mitigation. One chapter of the book [9] is dedicated to the power procurement problem from a large consumer's perspective, while another chapter discusses the case of a retailer in a medium-term horizon. In both cases, the electricity procurement problem is mathematically formulated as a multi-stage stochastic programming problem, where the evolution of the price is modelled as a stochastic process using a set of scenarios and the risk aversion is modelled through the CVaR. The work concludes that multi-stage stochastic programming appears to be an appropriate modelling framework to make procurement decisions under uncertainty, with the complex multi-stage stochastic model being translated into a tractable *mixed-integer linear programming* (MILP) problem.

Article [10] presents a novel approach based on the information gap decision theory to assess different procurement strategies for large consumers. The objective is not to minimise the procurement cost but rather to assess the risk aversion of procurement strategies with respect to the minimum achievable cost. The results suggest that strategies leading to a higher procurement cost are more robust and risk averse. Later on, the paper [11] proposes a robust optimisation methodology to solve the electricity procurement problem from the perspective of a retailer. A collection of robust mixed-integer linear programming problems is formulated, with the electricity price uncertainty being modelled by considering upper and lower limits for the energy prices rather than the forecast prices. Articles [12] and [13] present a stochastic optimisation approach relying on the integration of the paradigm of joint chance constraints and the CVaR risk measure to solve the electricity procurement problem from a consumer's perspective. The results for a real case study highlight the trade-off between risk and reliability by considering different levels of risk aversion. Paper [14] introduces another multi-stage stochastic programming model for the long-term electricity procurement task faced by a large consumer, where the complexity of the task is reduced by dividing a one-year planning into seasons. In this model, a season is represented by characteristic weeks and the seasonal demand is revealed at the beginning of each season. Article [15] presents a short-term decision-making model based on robust optimisation to help an electricity retailer in determining both the electricity procurement and its electricity retail price, so that profit is maximised. Two possibilities are offered to the retailer for its electricity procurement task: directly purchasing energy from generation companies or buying power on the spot market.

Paper [16] tackles a slightly different aspect of the electricity procurement problem as it studies how to size and use energy storage systems to minimise the procurement cost of electricity. The study focuses on short-term energy procurement by taking into consideration both the day-ahead market and the real-time market. Article [17] studies a multi-period power procurement problem in the specific context of smart-grid communities. The required energy can be obtained from both the day-ahead market, characterised by variable prices, and renewable energy sources, which are free but with uncertain supplies. To determine the optimal procurement amount, the authors adopt an approach based on dynamic programming, which has been proven to provide significant cost-savings. Finally, the article [18] introduces an agent-based two-stage trading model for direct electricity procurement of large consumers, which considers both the fairness and efficiency of direct energy procurement. The authors claim that this mechanism could offer more choices to both large consumers and generation companies that could also benefit from the reduction of the average market price.

To summarise, the algorithmic solutions to the electricity procurement problem found in the scientific literature are mostly based on stochastic programming, dynamic programming and optimisation techniques. Moreover, the sequential decision-making problem behind the electricity procurement task is formalised in numerous different ways with respect to the time horizon, the electricity power sources and markets, the energy consumption, among others. For this reason, a fair comparison between the different solutions is not feasible. Nevertheless, despite being sound and interesting works, the approaches from the literature are not well established within the industry. A potential explanation may be the tendency to have black box models, which are quite difficult for inexperienced employees to understand, interpret and monitor on a daily basis. Another observation about the scientific literature is the absence of methodologies based on advanced *machine learning* (ML) techniques. This research work attempts to fill this gap, by taking advantage of the promising results achieved by DL techniques in many fields of application. Finally, the key contribution of this research work can be summarised as follows: the design of an algorithmic solution taking advantage of advances in *artificial intelligence* (AI) to make sound and explainable trading decisions in the context of the long-term procurement of electricity.

2.3 Problem formalisation

In this section, the long-term electricity procurement problem is thoroughly presented and formalised. To begin with, it is assumed that the single supply source available to the agent, whether a retailer or a large consumer, is the *Calendar* (CAL) product from the Belgian forward market operated by *Ice Endex* (Belgian Power Base Futures). This yearly base load product is tradable up to three years ahead of the delivery period. For instance, the CAL 2018 product corresponds to the delivery of power during the entire year 2018, this energy being tradable between 2015 and 2017 included, as depicted in Figure 2.3. In fact, such an assumption is a slight simplification of the reality, where the agent may not only consider other products from the forward market but also the day-ahead and intraday markets if its consumption is not fully covered.

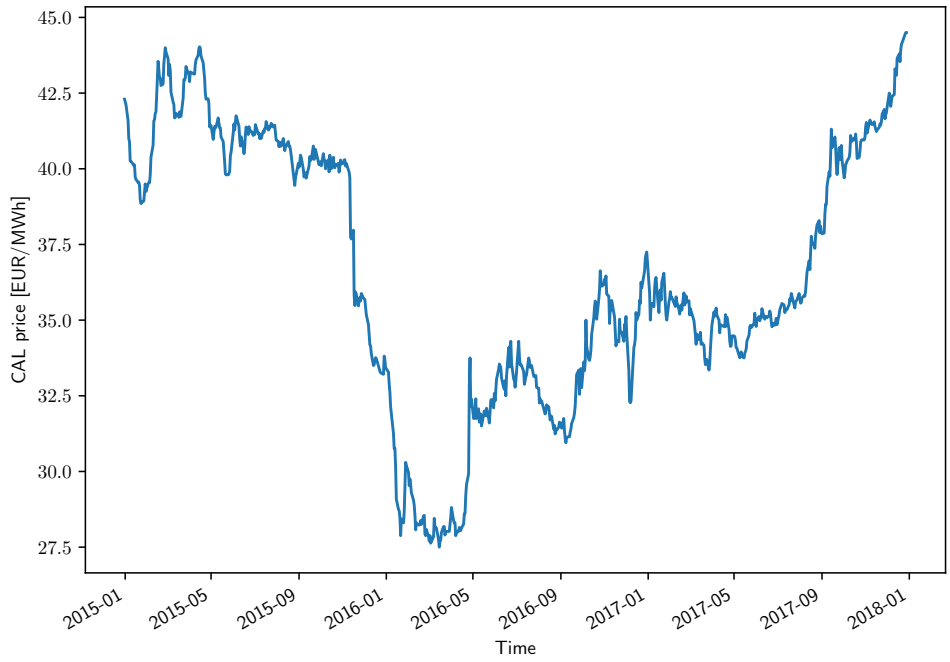


Figure 2.3: Evolution of the forward price related to the CAL 2018 product.

Prior to any trading activity, the electricity procurement problem requires to forecast the power consumption over the future period of interest. In this research work, the quantity of energy to be purchased over the procurement horizon is denoted Q. For the CAL product, this procurement horizon corresponds to a time period of three years and the quantity Q represents the consumption for one future year. In practice, the commodity exchanged in the forward markets is not a quantity of electricity, generally expressed in MWh, but rather an electrical power, commonly expressed in MW. In the following, this research work finds it more convenient to present its reasoning based on the quantity, knowing that a conversion to the power is easily performed since the delivery horizon is fixed (one year).

In the scope of this research work, the continuous trading timeline is discretised into a number of discrete time steps t of constant duration Δt. In this case, the agent is assumed to be able to make a single decision per trading day, meaning that Δt is equal to one full trading day. In the context of the long-term electricity procurement task, a trading or procurement strategy represents the set of rules adopted to make a decision. Mathematically, a procurement strategy is defined as a programmed policy $\pi : \mathcal{X} \to \mathcal{Y}$ which, based on some input information $x_t \in \mathcal{X}$ at time step t, outputs a trading decision $y_t \in \mathcal{Y}$ so as to maximise an objective criterion. The following subsections thoroughly detail the input, output and objective criterion defined for the electricity procurement problem studied.

2.3.1 Input of a procurement policy

Theoretically, the input of a procurement policy x_t at time step t has to ideally include every single piece of information that may potentially have an impact on the future electricity forward prices. However, a critical challenge associated with the power procurement problem is the unavailability of this information, which takes various forms and is both quantitative and qualitative. This situation leads to considerable uncertainty, with market directions that may be impossible to accurately explain and forecast. In this research work, the input x_t at time step t can be mathematically expressed as the following:

$$x_t = \{P_t,\ S_t\}, \tag{2.1}$$

where:

- $P_t = \{p_{t-\tau} | \tau = 1, ..., K\}$ is a time series composed of the last K prices from the CAL product traded, with $K \in \mathbb{N} \setminus \{0\}$ being a parameter,
- S_t is the state information of the trading agent, which is formally expressed as follows:

$$S_t = \{t,\ T,\ q_t,\ Q\}, \tag{2.2}$$

  with:

  - t being the current trading time step,
  - T being the total number of trading time steps over the procurement horizon,
  - q_t being the quantity of electricity already purchased by the agent at time step t,
  - Q being the total quantity to be purchased over the procurement horizon.

2.3.2 Output of a procurement policy

At each time step, the agent is assumed to have the decision whether to purchase electricity right now or to wait for a future opportunity. Therefore, the output of the procurement policy y_t at time step t is binary and can be mathematically expressed as the following:

$$y_t \in \{0,\ 1\}, \tag{2.3}$$

where $y_t = 0$ ($y_t = 1$) corresponds to the recommendation to hold/wait (buy), respectively. Whenever purchasing electricity, the agent is also required to specify the quantity of energy to be traded. In this research work, the volume contracted is assumed to be constant. The total quantity of electricity Q is split into $N \in \mathbb{N}\setminus\{0\}$ purchase operations of a fixed amount of energy $A = Q/N$. Consequently, the quantity of electricity purchased at each trading time step t would either be equal to 0 or A depending on the output y_t. However, this approach does not take into account the resolution of the market dQ, corresponding to the smallest block of electricity tradable. To address this issue, the quantity of energy Q is constrained to be a multiple of this market resolution dQ. Moreover, the parameter N is constrained to be such that the amount of electricity $A = Q/N$ is a multiple of the market resolution dQ.

An important constraint is imposed concerning the output of a procurement policy. By the end of the trading activity, meaning at the end of the trading horizon ($t = T$), the agent is required to have precisely purchased the quantity of electricity Q originally planned. In addition, it is assumed that the agent is not allowed to sell electricity in the forward market. If that operation was permitted, it would be a completely different decision-making problem closer to the topic studied in Chapter 3 of the doctoral thesis. Consequently, the agent is not allowed to buy electricity in excess of its planned consumption. Moreover, anticipation is necessary because the agent is only able purchase the amount of electricity A during a single time step. Formally, let $n_t = (Q - q_t)/A$ be the number of remaining purchase operations to be performed by the agent at time step t, this quantity must never exceed the number of remaining time steps $T - t$. For this reason, the output of a procurement policy may not always be freely selected. Eventually, this important constraint can be mathematically summarised as the following:

$$\sum_{t=0}^{T} y_t \ A = Q \ . \tag{2.4}$$

In order to realistically simulate the trading activity resulting from a procurement policy, the trading costs have to be seriously taken into consideration. This research work assumes that the only trading costs incurred by the agent are the transaction costs. As their name indicates, these costs occur each time a transaction (purchase) is performed. Therefore, the trading costs are modelled as a fixed fee C_F to be paid by the agent for each MWh of energy purchased in the forward markets.

Finally, making the hypothesis that the electricity is always successfully purchased by the agent, the state variable q_t is updated in line with the following equation:

$$q_{t+1} = q_t + y_t \ A \ . \tag{2.5}$$

2.3.3 Objective criterion

In the scope of the electricity procurement problem studied, the core objective is naturally the minimisation of the costs incurred for buying energy. However, such an intuitive goal lacks the proper consideration of the risk associated with the trading activity, which has to ideally be mitigated as well. In fact, there generally exists a trade-off between cost minimisation and risk mitigation, in accordance with the adage: with great risk comes great reward. Nevertheless, although risk mitigation is built in the proposed algorithmic solution, this research work adopts the costs minimisation alone as objective criterion, the inclusion of the risk being left as future work. Therefore, the quantity to be minimised is the total cost incurred by the agent at the end of the procurement horizon T, denoted c_T, which can be mathematically expressed as the following:

$$c_T = \sum_{t=0}^{T} y_t \ A \ (p_t + C_F) \ . \tag{2.6}$$

2.4 Algorithmic solution

In this section, a novel algorithmic solution denominated *Uniformity-based Procurement of Electricity* (UPE) is thoroughly presented to solve the long-term electricity procurement problem studied. The core idea behind this algorithm is to speed up or delay purchase operations with respect to a procurement strategy of reference, depending on the expected evolution of the market prices in the future. More precisely, the decision-making process is based on the coupling of two important pieces of information. Firstly, the identification of the dominant market direction. Secondly, the computation of the procurement uniformity level quantifying the deviation from a perfectly uniform procurement policy.

The first important component of the UPE algorithmic solution is the *forecaster* F whose responsibility is to predict the future dominant market trend, either upward or downward. In this research work, the market trend can be defined as the general direction in which the market price is going. Formally, at each trading time step t, the forecaster F takes as input a series of K previous forward prices P_t and outputs the predicted market trend f_t:

$$f_t = F(P_t), \tag{2.7}$$

where $f_t = 1$ ($f_t = -1$) corresponds to a forecast upward (downward) trend at time step t.

A market trend is a subjective notion that possesses different definitions in the scientific literature. For instance, some may argue that a sudden decrease in price during a week is a new downward trend, when others view this behaviour as a temporary phenomenon within a more global upward trend lasting for months. Actually, both views may be right depending on the time horizon taken into consideration. In order to eliminate any ambiguity, this research work adopts the following methodology to elaborate an objective mathematical definition. A lag-free low-pass filtering operation of large order k, typically several weeks, is applied to the market price signal. The resulting smoothed price signal $\bar{p}_t$ at time step t can be mathematically expressed as the following:

$$\bar{p}_t = \frac{1}{2k+1} \sum_{\tau=t-k}^{t+k} p_\tau \ . \tag{2.8}$$

As an illustration, the result of this lag-free low-pass filtering operation with $k = 25$ is depicted in Figure 2.4 for the CAL 2018 product. Based on this new information, the market trend f_t at time step t is derived from the comparison of the two consecutive smoothed prices $\bar{p}_t$ and $\bar{p}_{t-1}$. More precisely, an upward trend $\hat{f}_t = 1$ is designated when $\bar{p}_t \geq \bar{p}_{t-1}$, and a downward trend $\hat{f}_t = -1$ is specified when $\bar{p}_t < \bar{p}_{t-1}$, with $\hat{f}_t$ representing the market trend label at time step t. Although this rigorous mathematical definition of a market trend is intuitive and convenient, it is not perfect and there exist other relevant definitions that could be investigated as future work. For instance, the market trend may be defined as the slope (sign) of the straight line produced by a linear regression operation on price data over a certain time period, as illustrated in Figure 2.5. Finally, despite being subjective, human annotations could alternatively be considered as well.

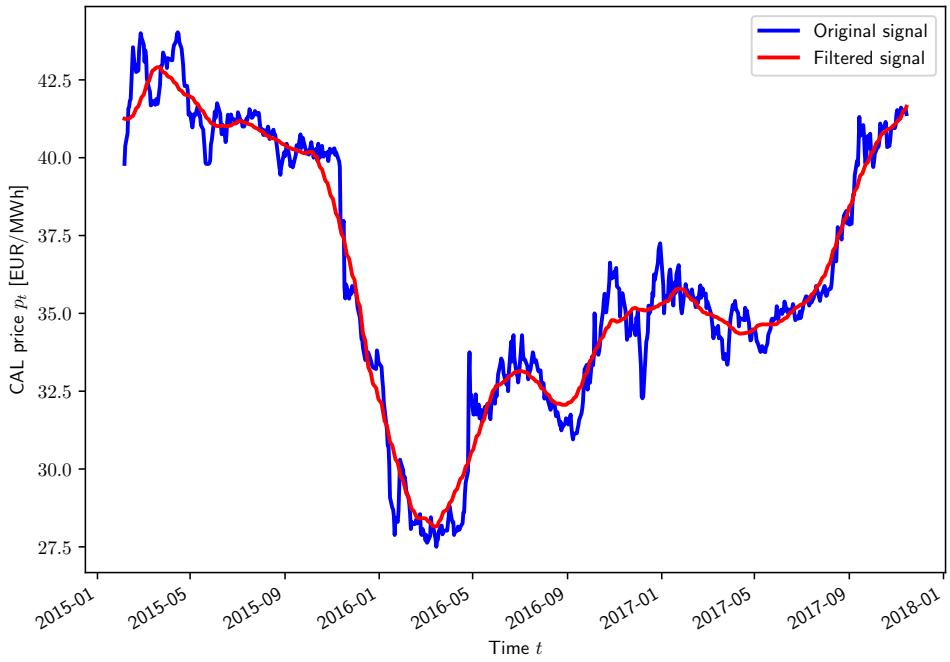


Figure 2.4: Lag-free low-pass filtering operation performed on the CAL 2018 product.

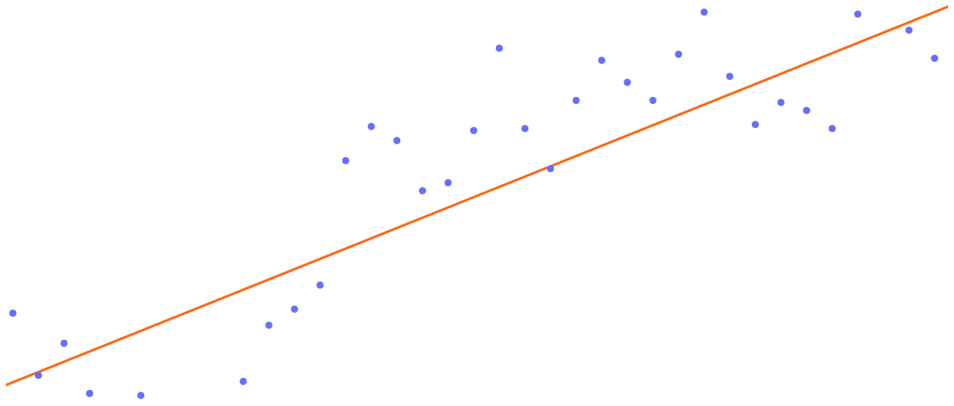


Figure 2.5: Linear regression (orange line) performed on a fictive price signal (blue dots).

The second important component of the UPE algorithm is the concept of *procurement uniformity deviation*. At its name indicates, this quantitative indicator compares the trading activity of the agent to a strategy of reference named *perfectly uniform procurement policy*. This particular procurement strategy consists in purchasing the same amount of electricity $A_u = Q/T$ at each trading time step t over the entire procurement horizon T. Despite not being generally feasible in practice because of the market resolution dQ, this procurement policy is an interesting candidate for comparison purposes since it achieves the average power price and spreads the trading risk over the full procurement horizon. Therefore, this research work introduces the procurement uniformity level $u_t \in [-1,\ 1]$ that quantifies the deviation from such a perfectly uniform procurement policy of reference:

$$u_t = \frac{T-t}{T} - \frac{Q-q_t}{Q}\ . \tag{2.9}$$

Three cases arise depending on the value of the procurement uniformity level u_t:

- **Null value ($u_t = 0$).** The agent has purchased at time step t a quantity of electricity equal to the amount of energy that a perfectly uniform procurement policy would have already bought at that time.
- **Positive value ($u_t \in\ ]0,\ 1]$).** The trading agent leads at time step t compared to a perfectly uniform procurement policy, meaning that it has already purchased a larger quantity of electricity.
- **Negative value ($u_t \in\ [-1,\ 0[$).** The trading agent lags behind at time step t compared to a perfectly uniform procurement policy, meaning that it has already purchased a smaller quantity of electricity.

The decision-making process is based on the comparison of the procurement uniformity level u_t with two trigger values u^- and u^+. As indicated in Equation (2.9), the quantity u_t decreases over time when the agent holds with some purchase operations remaining. The idea of the UPE algorithm is to issue a new purchase operation when this indicator hits the trigger value associated with the predicted trend, u^+ for upward and u^- for downward. Therefore, the triggers values represent how long the agent is willing to wait for each market trend. These are key parameters of the algorithmic solution, to be set by the agent according to its expectations regarding the market dynamics and its sensitivity to the trading risk.

More precisely, the UPE algorithm proceeds as follows to generate trading recommendations on the basis of both the market trend forecasts and the procurement uniformity level. While a downward market trend is predicted by the forecaster F ($f_t = -1$), meaning that the prices are expected to decrease in the future, the agent is advised to wait as long as possible. In this case, the purchase operation is recommended once the procurement uniformity level u_t goes below the trigger value u^-, in order to not lack too much behind the reference policy. On the contrary, if an upward market trend is likely to happen according to the forecaster F ($f_t = 1$), the agent is advised to directly buy as much as possible. In this situation, a new purchase operation is suggested as soon as the procurement uniformity level u_t goes below the trigger value u^+, so as not to be too far ahead of the policy of reference.

For the sake of completeness and clarity, Algorithm 1 details the decision-making process behind the UPE algorithm for a single time step t. In addition, a graphical illustration (flowchart) of the decision-making is presented in Figure 2.6. Line 2 of Algorithm 1 indicates that the first operation of the UPE algorithm at each time step t is to potentially retrain or retune the forecaster on the basis of the newly available data. This optional operation consists in updating the forecasting model F at a certain frequency (daily, weekly, monthly) to take advantage of the latest data available. Indeed, at each time step t, the forecaster F can be trained (tuned) using all the data available up to time step t. Therefore, the training dataset is continuously growing over time, and it could be interesting to retrain (retune) the forecasting model as new data becomes available. In practice, this particular technique may be helpful to ensure the adaptability of the UPE algorithm to changing market conditions. Nevertheless, this approach is left as future work, because of the important computational cost associated. In the following, the forecasting model F is trained (tuned) once and remains fixed over the entire procurement horizon.

Algorithm 1 Decision-making process behind the UPE algorithm for a single time step t

1: **Inputs:** procurement policy input x_t, formerly trained/tuned forecaster F, triggers values u^- and u^+.
2: If applicable, retrain/retune the forecasting model F according to the newly available data.
3: Generate a new market trend forecast $f_t = F(P_t)$.
4: Compute the procurement uniformity $u_t = \frac{T-t}{T} - \frac{Q-q_t}{Q}$.
5: **if** $f_t = 1$ **and** $u_t < u^+$ **then**
6: Make the trading decision to purchase electricity: $y_t = 1$.
7: **else if** $f_t = -1$ **and** $u_t < u^-$ **then**
8: Make the trading decision to purchase electricity: $y_t = 1$.
9: **else**
10: Make the trading decision to hold/wait: $y_t = 0$.
11: **end if**
12: **return** y_t

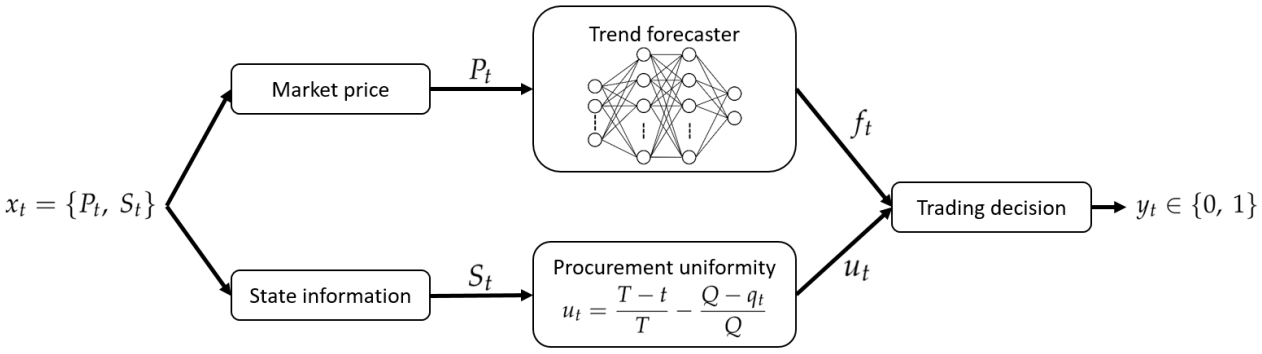


Figure 2.6: Flowchart of the decision-making process behind the UPE algorithm.

In this research work, two different forecasting models are adopted to predict the market trends. Thoroughly described hereafter, they are respectively denominated *basic forecaster* and *DL forecaster*.

2.4.1 Basic forecaster

In finance, a popular approach to acquire insights about the market trend from past data consists in comparing two moving averages of different window lengths [19]. The core idea is to assess how the more recent prices represented by the shorter moving average evolve with respect to the older prices described by the longer moving average. Both windows lengths L^{short} and L^{long} are parameters to be tuned, with typical values being several weeks or months. The moving average of window length L at time step t is expressed as follows:

$$M_t(L) = \frac{1}{L} \sum_{\tau=t-L}^{t-1} p_\tau \ . \tag{2.10}$$

With such a definition, an upward trend ($f_t = 1$) is naturally expected when the shorter moving average is greater than the longer one: $M_t(L^{short}) \geq M_t(L^{long})$. On the contrary, a downward trend ($f_t = -1$) is awaited when the shorter moving average is lesser than the longer one: $M_t(L^{short}) < M_t(L^{long})$. This conventional approach is adopted for the first forecasting model F^{MA} tested. The UPE algorithm using this forecaster is denominated *Uniformity-based Procurement of Electricity with Moving Averages* (UPE-MA).

2.4.2 DL forecaster

A more advanced approach inspired by recent successes in DL is investigated for the second forecasting model. This forecaster, denoted F^{DL}, consists of a *residual deep neural network* (ResDNN). More precisely, this model is a *multilayer perceptron* (feedforward artificial NN) equipped with *residual connections* [20]. Also called skip connections, the latter have been proven to significantly improve the training of DNNs by avoiding the problem of vanishing gradients as well as mitigating the degradation problem (accuracy saturation). A typical residual block is depicted graphically in Figure 2.7. The ResDNN architecture is composed of several residual blocks sequentially connected. The number of residual blocks n_b, the number of hidden layers n_l and the number of neurons per layer n_n are parameters of the DL model to be tuned. Regarding the choice of the activation functions, *leaky rectified linear unit* (Leaky ReLU), introduced in the article [21], are selected for the hidden layers. This popular activation function is mathematically expressed as follows:

$$f(x) = \begin{cases} x & \text{if } x > 0, \\ 0.01x & \text{otherwise.} \end{cases} \tag{2.11}$$

Because the market trend forecasting task is a classification problem, a *softmax* activation function is selected for the output layer. This enables to return valid probabilities for each market trend, as explained in the book [22]. Naturally, the forecaster F^{DL} outputs the market trend with the greatest probability. Formally, the softmax activation function takes as input a vector of real numbers $\boldsymbol{X} = (x_1, ..., x_J) \in \mathbb{R}^J$ and outputs a vector of J real numbers bounded between 0 and 1 representing probabilities:

$$S(\boldsymbol{X})_i = \frac{e^{x_i}}{\sum_{j=1}^{J} e^{x_j}} \qquad \forall i \in \{1, ..., J\}. \tag{2.12}$$

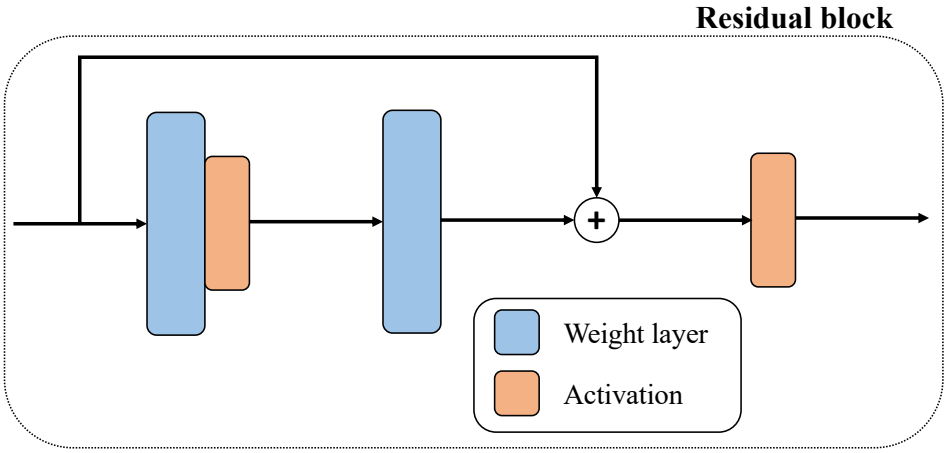


Figure 2.7: Illustration of a typical residual block.

The choice of the loss to be minimised is also conditioned by the fact that the market trend forecasting task is a classification problem. The *cross-entropy* loss, also referred to as logarithmic loss and inspired from the article [23], is selected for that purpose. This particular loss is mathematically expressed as the following:

$$\mathcal{L}(\theta) = \frac{1}{B} \sum_{b=1}^{B} -\log(p(y_b = \hat{y}_b | x_b, \theta)), \tag{2.13}$$

where:

- B is the batch size,
- x denotes the input of the DNN,
- y denotes the output of the DNN,
- $\hat{y}$ represents the classification labels,
- θ represents the parameters of the DNN (Weights and biases),
- $p(\star)$ represents the probability of the event $\star$.

The training of the DL forecasting model is performed using the *ADAM* optimiser, which is introduced in the article [24]. Moreover, in order to prevent overfitting of the ML model, both *dropout* and *L2 regularisation* techniques are included within the forecasting model. Finally, the inputs of the DL model are properly normalised, and some data augmentation techniques for time series are taken into consideration. For more details about the various ML and DL techniques previously mentioned, please refer to the article [25] and the book [22]. As an illustration, Figure 2.8 depicts the ResDNN architecture composing the DL forecaster, with the weight layers being fully connected (dense) layers in this case. In the following, the UPE algorithm operating the forecaster F^{DL} is denominated *Uniformity-based Procurement of Electricity with Deep Learning* (UPE-DL).

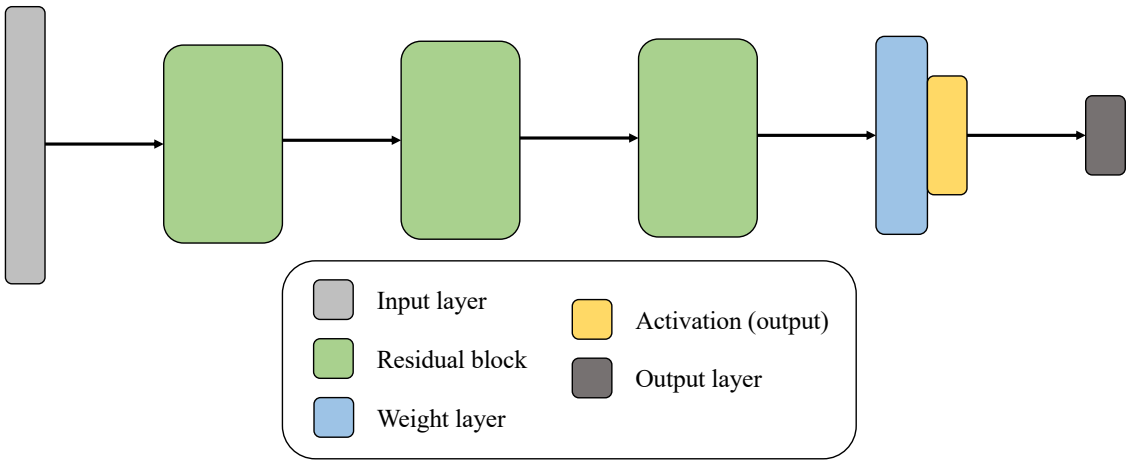


Figure 2.8: Illustration of the ResDNN architecture composing the DL forecaster.

2.5 Performance assessment methodology

In this section, the performance assessment methodology adopted by this research work to evaluate the performance of a long-term electricity procurement policy is presented. The case study selected consists of multiple CAL products over a time period of eight years, ranging from CAL 2012 to CAL 2019. A single CAL product is approximately composed of 750 daily power prices. If missing samples or abnormal values are detected within the dataset provided by Ice Endex, new data are artificially generated by linear interpolation or extrapolation as replacements for the problematic samples. As an illustration, Figure 2.9 depicts graphically the full set of data composing this case study. It can be clearly observed from this plot that the procurement policy assessed will be confronted to various market phenomena, including both upward and downward trends together with diverse levels of volatility. Finally, in order to present meaningful results, the training and test sets are naturally separated. Both the tuning of the parameters and the training of the DL forecasting model are performed on the basis of the CAL product three years prior to the one actually tested, so that no data are shared between the training and test sets. For instance, the training of a procurement policy for the CAL 2018 product, with electricity purchased between 2015 and 2017, is performed with data from the CAL 2015 product, for which energy is bought between 2012 and 2014.

For comparison purposes, two basic benchmark procurement strategies are included within the performance assessment methodology. The first one is named *Naive Balanced Electricity Procurement* (NBEP). This strategy simply consists in spreading the N purchase operations evenly over the time horizon. More precisely, the procurement horizon is divided into N intervals of identical durations and a single purchase operation is executed in the middle of each interval. In fact, this basic strategy is the realistic version of the perfectly uniform procurement policy, taking into consideration the market resolution. The second benchmark strategy is denominated *Electricity Procurement with Moving Averages* (EPMA), which is an adaptation of the moving averages trend following strategy to the electricity procurement task. More details about this popular algorithmic trading approach in finance can be found

in the book [19]. This procurement strategy is primarily based on the basic forecaster from Section 2.4.1, with two moving averages over different time periods for predicting the future market trend. A purchase operation is simply triggered each time a new upward trend is detected, occurring when the shorter moving average $M_t(L^{short})$ crosses and becomes greater than the longer moving average $M_t(L^{long})$. If the number of purchase operations performed is smaller than N by the end of the procurement horizon, the remaining ones are executed during the last trading time steps.

As previously explained in Section 2.3.3, the ultimate objective of the procurement policy is the minimisation of the total cost c_T. In order to improve the readability of the results, the quantitative performance indicator $C = c_T/Q$, representing the average price expressed in €/MWh at which the electricity is purchased, is adopted instead. Additionally, several procurement strategies of reference achieving particular values for the performance indicator C are also taken into consideration for comparison purposes. Firstly, the optimal and worst procurement policies respectively achieving the minimum and maximum costs are examined. These unrealistic strategies are trivially derived once the evolution of the power price over the entire procurement horizon is known. Secondly, the average electricity price achieved by the perfectly uniform procurement strategy of reference is computed, even though this policy is generally not feasible in practice because of the market resolution dQ. Lastly, the UPE algorithm equipped with a perfect forecaster achieving 100% accuracy, meaning always correctly predicting the market trend labels $\hat{f}_t$ defined in Section 2.4, is investigated under the denomination UPE-F.

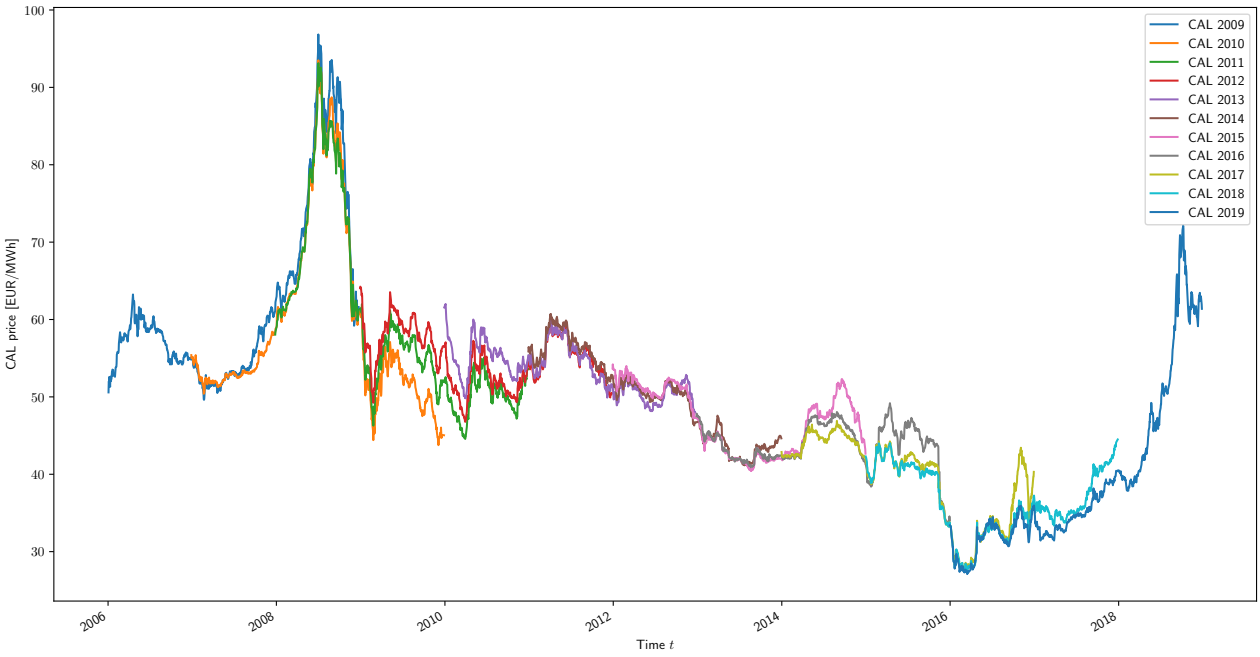


Figure 2.9: Illustration of the full set of CAL data composing the case study.

2.6 Results

Before getting into the analysis of the results, several elements are specified to ensure the reproducibility of the experiments. Firstly, the hyperparameters adopted in the simulations are revealed in Table 2.1. Secondly, the data composing the case study, which are depicted in Figure 2.9, have been provided by Ice Endex [26]. The dataset is not freely and publicly available, but can be purchased or may potentially be requested for research purposes. Finally, the complete experimental code of the algorithmic solution discussed can not be publicly shared because the present research has been conducted in the scope of a partnership with a company from the private sector.

To begin with, following the performance assessment methodology previously described, Table 2.2 summarises the results achieved by the novel procurement policies introduced (UPE-MA, UPE-DL) alongside the benchmark procurement strategies (NBEP, EPMA) and references (Min, Mean, Max, UPE-F).

Table 2.1: Hyperparameters adopted within the experiments.

Name	**Symbol**	**Value**
Quantity of electricity to buy [MWh]	Q	10^6
Number of purchase operations	N	10
Length of the input time series [days]	K	50
Order of the low-pass filtering operation	k	25
Procurement uniformity trigger −	u^-	-0.3
Procurement uniformity trigger +	u^+	0
Number of training epochs (DL model)	e	10^4
Learning rate (DL model)	l_r	10^{-5}
Dropout probability (DL model)	D_p	0.2
L2 regularisation factor (DL model)	L_2	10^{-3}
Number of residual blocks (ResDNN)	n_b	3
Number of layers per residual block (ResDNN)	n_l	2
Number of neurons per layer (ResDNN)	n_n	256

Table 2.2: Comparison of the electricity cost C [€/MWh] achieved by the different procurement policies.

CAL product	**Procurement policies**				**References**			
	NBEP	**EPMA**	**UPE-MA**	**UPE-DL**	**Min**	**Mean**	**Max**	**UPE-F**
2012	54.903	52.854	56.076	**52.032**	47.289	55.005	63.319	52.879
2013	53.564	**52.566**	52.826	53.548	48.239	53.614	60.638	52.400
2014	49.387	51.346	50.063	**48.762**	41.233	50.234	60.279	49.036
2015	45.834	**44.402**	47.400	47.614	40.580	47.043	53.713	46.230
2016	43.613	**43.038**	43.927	43.374	34.077	43.912	48.631	41.632
2017	38.887	**38.707**	39.940	39.477	27.814	39.449	46.435	37.756
2018	36.357	38.666	**34.537**	36.206	27.727	36.440	43.852	35.443
2019	42.532	48.526	37.749	**37.501**	27.310	39.017	70.693	37.335
Average	45.635	46.263	45.315	**44.814**	36.784	45.589	55.945	44.089

Average performance. Taking into consideration solely the average performance that is presented on the last line of Table 2.2, the two variants of the novel UPE algorithm clearly outperform both benchmark procurement strategies. Moreover, the UPE-MA and UPE-DL procurement policies respectively perform 0.6% and 1.7% better than the perfectly uniform procurement strategy of reference (column *Mean* in Table 2.2). More precisely, the former percentage represents the relative reduction in electricity costs C to be expected with respect to the situation of purchasing at the average power price over the procurement horizon. Even though the improvement in performance achieved by the novel algorithmic solution may seem to be quite limited at first glance, it actually corresponds to a comfortable annual saving of hundreds of thousands of euros for large electricity consumers/retailers. For instance, the UPE-DL procurement policy achieves an expected yearly saving of €775,000 with respect to the perfectly uniform procurement strategy for an annual consumption of 1 TWh of energy. Additionally, the maximum achievable reduction in costs is naturally not 100%, but rather 19.31% for this case study (column *Min* in Table 2.2), which is not even realistically feasible in practice. The promising results achieved indicate that the novel algorithmic solution is able to identify and take advantage of certain market phenomena. Moreover, the experiments suggest that the DL forecasting model F^{DL} outputs more accurate and relevant market trend predictions compared to the more conventional forecaster F^{MA}, the accuracy of a forecaster being defined as the number of correct predictions $f_t = \hat{f}_t$ compared to the total number of predictions. On average, the performance of the UPE procurement policy is expected to increase as the forecasting model becomes more accurate while staying consistent, as proven by the UPE-F reference strategy that is characterised by a 100% accuracy.

Results variance. In addition to the expected performance, the variance of the results achieved by the novel algorithmic solution has to be rigorously analysed. Indeed, such an assessment provides valuable information on the robustness and stability of a procurement strategy, which is key for its application in real life. First of all, it has to be highlighted that the mean electricity price reported in Table 2.2 significantly varies over the years. Therefore, the variance in performance has to be discussed after subtracting the mean electricity prices from the electricity costs C achieved by a procurement policy. Following this methodology, the results variance substantially differs depending on the procurement policy analysed. On the one hand, the EPMA strategy achieves the best performance for half the years of the case study but totally fails the CAL 2019 product, because of a flaw in its design further discussed later. On the other hand, the NBEP strategy is never the best procurement policy but achieves a lower variance without any significant failure, as expected. As far as the UPE algorithm is concerned, both variants deliver consistent results which are at least comparable and generally better than the mean electricity price of reference. This consistency throughout the years demonstrates the stability and robustness of the proposed algorithmic solution, with positive results achieved whatever the price dynamics. In the scope of the long-term electricity procurement task studied, this is particularly important because of the subtiantial uncertainty characterising the sequential decision-making problem.

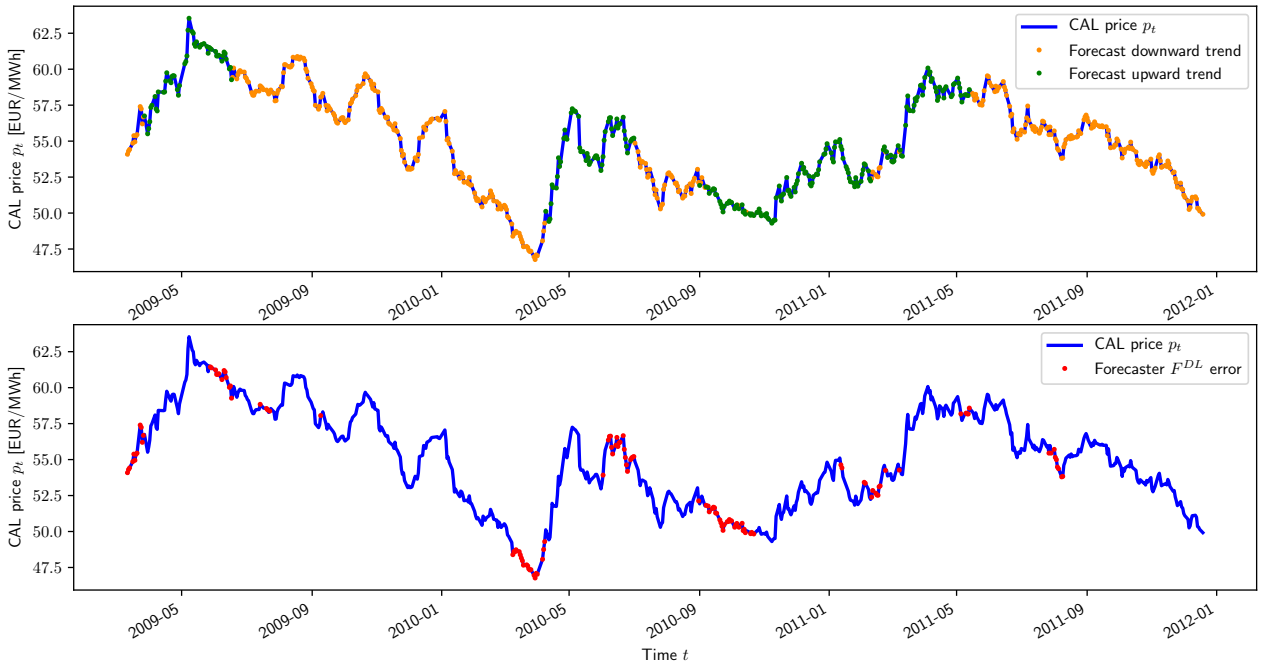


Figure 2.10: Predicted market trends (top) and forecasting errors (bottom) of the forecaster F^{DL} in the context of the CAL 2012 product.

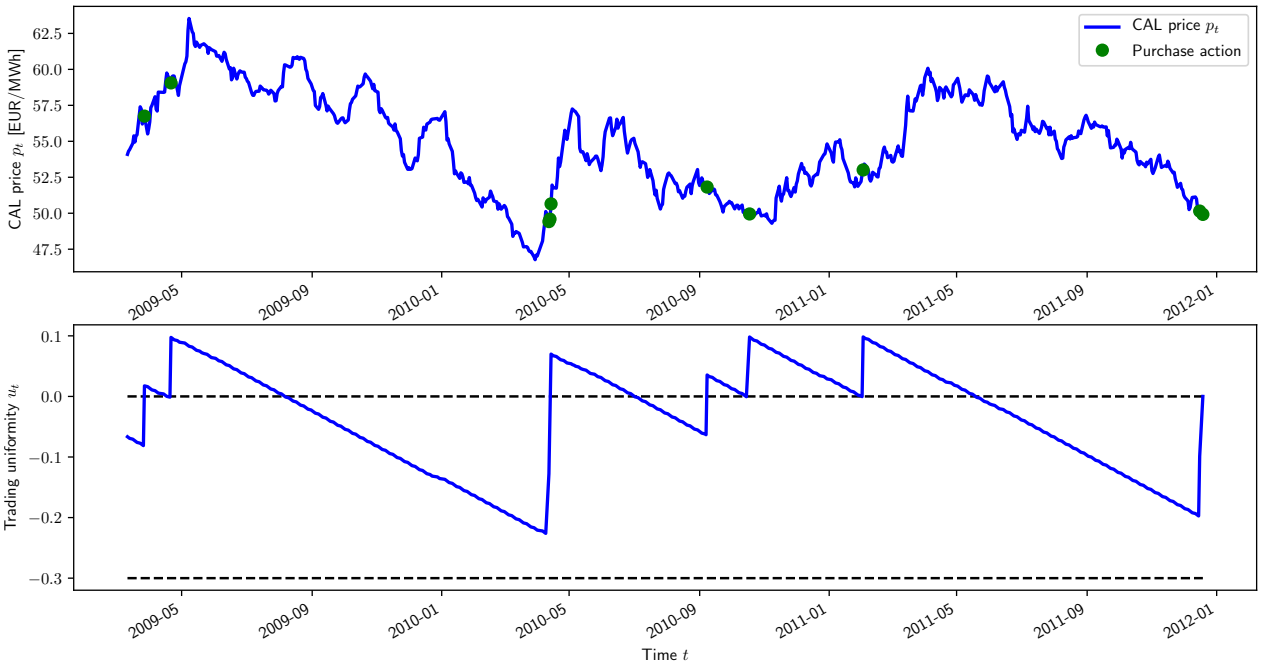


Figure 2.11: Purchase operations performed by the UPE-DL algorithm (top) alongside the evolution of the procurement uniformity u_t (bottom) in the context of the the CAL 2012 product.

Representative application of the UPE-DL algorithm. Figures 2.10 and 2.11 do illustrate the application of the UPE-DL algorithmic solution in the context of the CAL 2012 product. Firstly, Figure 2.10 presents the market trend predictions f_t outputted by the forecaster F^{DL} alongside the market prices p_t in the upper plot, together with the forecasting errors $f_t \neq \hat{f}_t$ in the bottom plot. For this particular year, the DL forecasting model achieves very encouraging results, with an accuracy of approximately 80% together with convenient forecasts. Indeed, the predictions do not incorrectly oscillate between the two market trends during time periods of pronounced volatility, a behaviour which could significantly harm the performance of the UPE-DL algorithm. For instance, in the case of a major downward market trend with several temporary limited rebounds, erroneous forecasts of upward trends would lead to the trigger of some purchase operations prematurely. Secondly, Figure 2.11 presents the evolution of the market price p_t alongside the purchase decisions $y_t = 1$ in the upper plot, together with the associated procurement uniformity level u_t in the bottom plot. This illustration demonstrates the ability of the UPE-DL procurement policy to delay buying operations when the market price is expected to decrease in the future, so that purchases are completed close to local price minima. Moreover, the bottom plot exhibits how the two procurement uniformity thresholds u^- and u^+ may actually be viewed as a risk mitigation mechanism built into the UPE algorithm. Finally, Figures 2.10 and 2.11 highlight the interpretability of the trading decisions recommended by the UPE-DL algorithm. Having a decision-making process which is explainable and completely transparent to the human supervisor significantly eases the monitoring of the procurement strategy and improves its reliability, a critical feature for its application in the industry.

Sensitivity analysis. The long-term electricity procurement task is dependent on the number of purchase operations N to be performed over the procurement horizon. Contrarily to other parameters that are freely tuned, this quantity is constrained because of the market resolution dQ. For this reason, the sensitivity of the performance achieved by a procurement strategy to the number of purchase operations N has to be analysed. Figure 2.12 illustrates the impact of the constrained parameter N on the expected electricity cost C achieved by different procurement policies. Firstly, it can be clearly observed that the number of purchase operations has to naturally be large enough to observe the true behaviour of a procurement strategy, in this case $N \geq 5$ typically. Otherwise, a completely different situation arises where the results are generally a matter of luck. Secondly, the UPE-DL algorithm remains the leading procurement policy, whatever the value of the parameter N. Additionally, the orange curve (UPE-MA) is roughly shifted up compared to the blue one (UPE-DL), which is another indication that the DL forecasting model generates better market trend predictions. Thirdly, regarding the benchmark procurement strategies, NBEP is, by design, the most resilient policy to changes in the number of buying operations. As expected, its performance converges towards the mean electricity price when this parameter increases. On the contrary, the EPMA strategy has a real flaw in its design since it does not monitor the number of purchase operations. This makes the procurement policy more sensitive to the constrained parameter N, and therefore significantly less reliable.

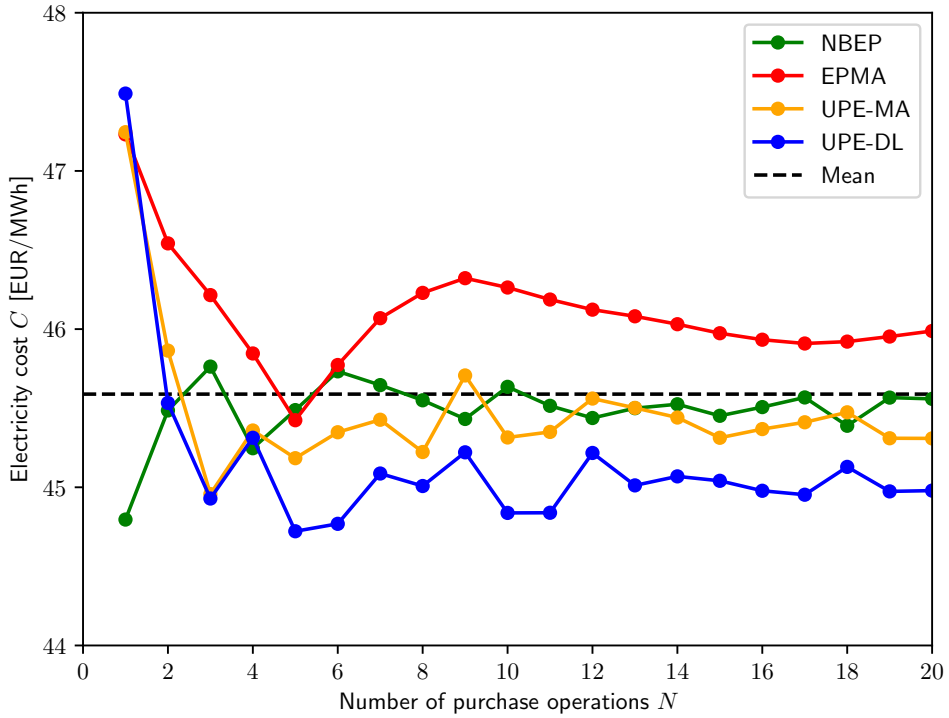


Figure 2.12: Evolution of the average performance (electricity cost C) achieved by the different procurement policies with respect to the number of purchase operations N.

2.7 Discussion

On the basis of the results previously analysed, this section discusses the main advantages and limitations of the proposed approach. Firstly, the novel algorithmic solution introduced successfully takes advantage of the impressive forecasting capabilities of DL techniques. With future innovations in this area, the performance of the UPE algorithm is expected to further improve over time. To the authors' knowledge, this is the first research work considering advanced AI techniques to solve the sequential decision-making problem behind the power procurement task, with existing solutions being mostly based on stochastic programming and optimisation methods. Secondly, the UPE algorithm presents the key advantage of producing trading recommendations which are easier to interpret, a particularly important feature for its application in the industry. This is achieved by coupling the market trend forecasting with the procurement uniformity level within the decision-making process. Lastly, a risk mitigation mechanism is included within the procurement policy, through the limitation of the deviation from a perfectly uniform procurement policy (triggers u^- and u^+). This makes the algorithmic solution more robust to exceptional events such as economic crises, during which the forecasting model may potentially be inaccurate. With appropriate tuning of the two parameters u^- and u^+, performance collapses can truly be prevented, or at the very least mitigated, due to the constrained deviation with respect to a perfectly uniform procurement policy of reference.

Concerning the weaknesses, the proposed approach suffers from two main limitations. Firstly, the quality of the trading recommendations generated by the UPE algorithm is strongly dependent on the performance achieved by the market trend forecasting model. Nevertheless, if a complex DL model is used for that purpose, the forecaster F becomes a black box whose output is quite difficult to interpret. This may potentially have a negative impact on the explainability of the trading recommendations. Secondly, as previously explained, the deviation from a perfectly uniform procurement policy of reference is constrained according to the two procurement uniformity triggers u^- and u^+. Despite being a useful mechanism to effectively mitigate the risk, it may also limit the profit (cost reduction) when the forecasting model is particularly accurate. Therefore, this mechanism may be viewed as both a strength and a weakness, depending on the market dynamics and on the performance of the market trend forecaster. In addition, the tuning of the two parameters u^- and u^+ is an appreciable degree of freedom, but which may be quite challenging to perform.

2.8 Future work

Even though the novel algorithmic solution presented does achieve promising results, there is still room for improvements. In the following, several avenues are suggested as future work. Firstly, as previously explained in Section 2.3.3, the risk associated with the trading activity of the agent in the scope of the long-term electricity procurement task has to be rigorously defined and mitigated. This exercise is expected to be complicated because the trading risk may have several different definitions for this particular sequential decision-making problem. Once a proper mathematical definition is available, the risk has to be included within the objective criterion of procurement policies alongside cost minimisation.

The second idea promoted concerns the dataset used to train the market trend forecasting model. First, the performance of the forecaster could undoubtedly be improved by taking into consideration additional information within the input space. Indeed, the procurement policy input x_t is not sufficient to accurately explain most financial phenomena occurring within the forward markets. The forecasting model could really benefit from supplementary information including the price evolution of correlated commodities, some macroeconomic data, and perhaps most importantly the news. In fact, the price from the forward market reflects the expectation of the market participants regarding the future average price from the day-ahead market over a certain period of time. Therefore, having access to the general feeling of these market participants about this subject is particularly important for accurately forecasting the future price direction. Although it is impossible to directly retrieve this information, monitoring the news that may have a substantial impact on the perception of these market participants is feasible. Second, the training set could be extended, since a sufficiently large dataset is required to effectively train complex DL models. In this research work, the amount of training data is fairly limited compared to the complexity of the forecasting task. Therefore, not only could the training set range over a longer period of time, but data augmentation techniques relevant to financial time series could also be taken into consideration, in order to generate additional data samples. Last, updating the forecasting model as new data becomes available is key to adapt to changing market dynamics.

Thirdly, the forecasting model could potentially benefit from more advanced techniques promoted by the DL research community in the scope of various forecasting problems. In addition, other state-of-the-art forecasting approaches may also be considered [27, 28]. The first architecture highlighted is the *residual convolutional neural network* (ResCNN), which is similar to the ResDNN architecture except for the fully connected layers within the residual blocks that are replaced by 1D convolutional layers with batch normalisation [29]. As an illustration, the convolutional version of the residual block proposed is depicted graphically in Figure 2.13. A second DL architecture to further explore is the *long short-term memory* (LTSM), a recurrent neural network which has been demonstrated to better deal with the vanishing gradient problem generally encountered when training such recurrent models [30]. Theoretically, this architecture excels at modelling sequential data, including time series. This is the main reason for considering such a recurrent model in the scope of the market trend forecasting task. An illustration of the LSTM network adapted to this forecasting problem is provided in Figure 2.14. The last architecture suggested for the market trend forecasting task is the *transformer*. This particular DL model is composed of an encoder and a decoder, and leverages the attention mechanism [31]. The latter technique is a key step towards the valuable interpretability of complex DL models, by giving access to the focus (attention) of the black box model among its input features to generate its output. The transformer has already been demonstrated to be particularly successful in the field of *natural language processing* (NLP). More recently, research works highlighted the potential of this innovative architecture for time series forecasting [32, 33]. In particular, the *temporal fusion transformer* (TFT) [33] is a promising model to deal with such complex problems. For illustration purposes, the general transformer architecture adapted to the market trend forecasting task at hand is graphically depicted from a high-level perspective in Figure 2.15. Moreover, the attention mechanism would be particularly helpful to further improve the interpretability of the UPE algorithm introduced.

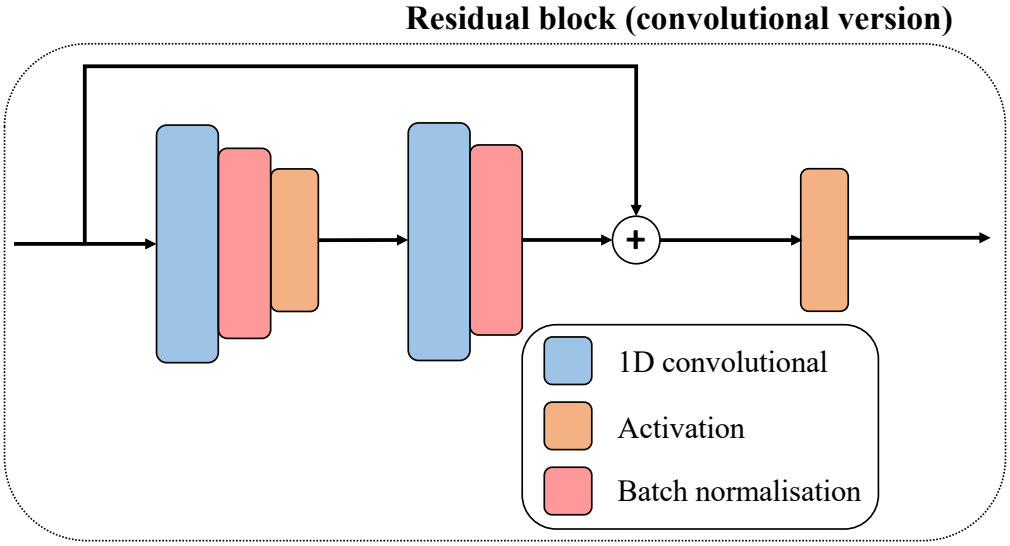


Figure 2.13: Illustration of the convolutional version of the residual block previously presented.

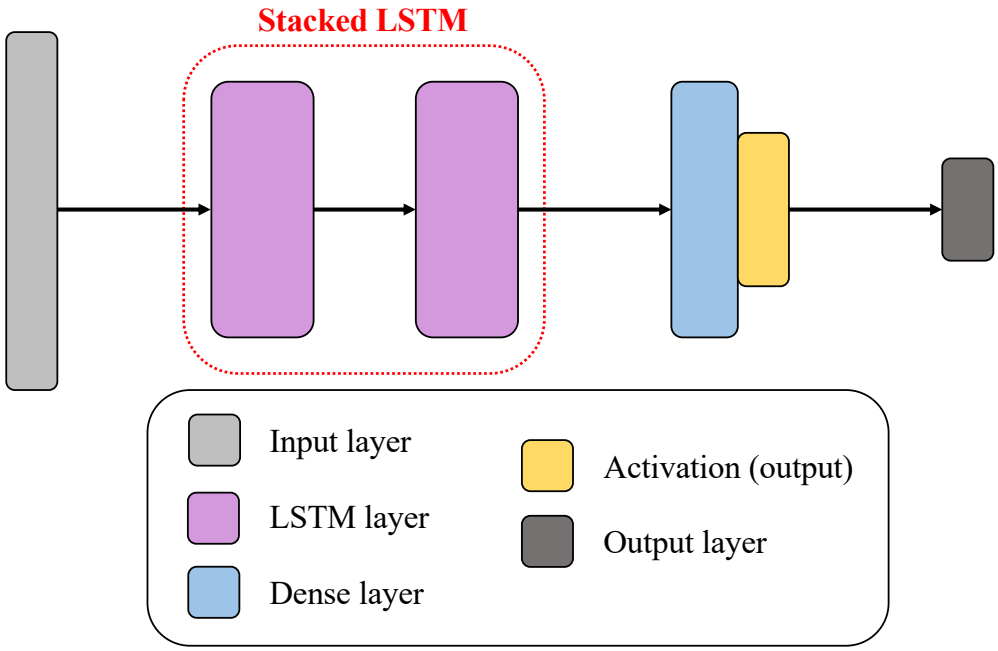


Figure 2.14: Illustration of the suggested LSTM architecture adapted to market trend forecasting.

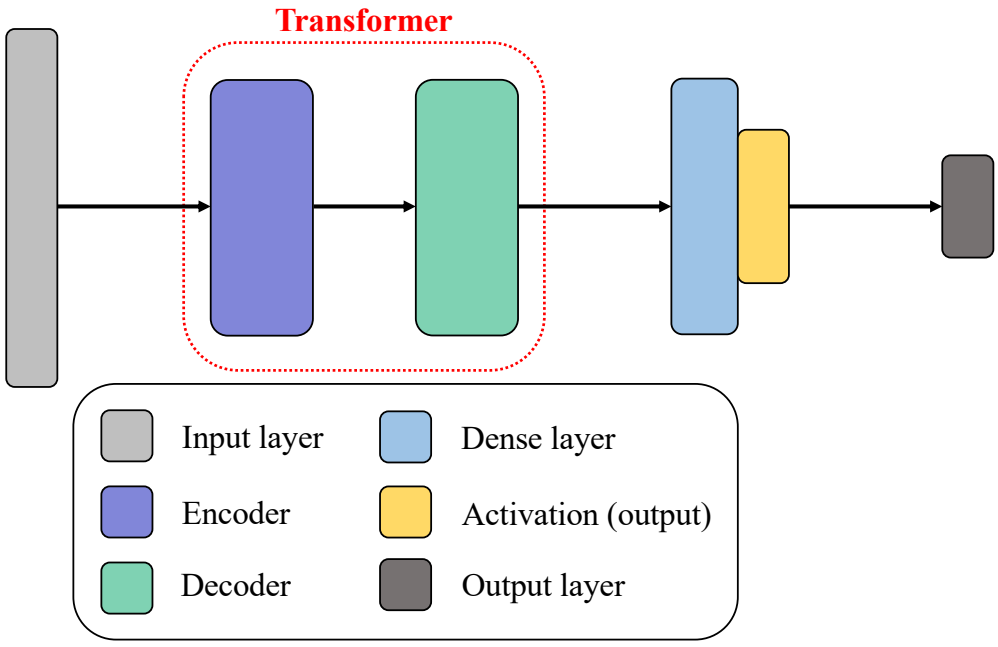


Figure 2.15: Illustration of the proposed transformer architecture adapted to market trend forecasting.

Lastly, an interesting alternative approach to solve the complex sequential decision-making problem studied is the subfield of AI called *deep reinforcement learning* (DRL). In short, this methodology is concerned with the learning process of an agent (i) sequentially interacting with an unknown environment (ii) aiming to maximise its cumulative rewards (iii) using DL techniques to generalise the information acquired from the interaction with the environment. The multiple recent successes of DRL solutions in various fields such as gaming and robotics highlight the potential of this novel approach. Applying such an advanced AI method to the long-term electricity procurement problem will be particularly challenging, but is expected to lead to interesting decision-making policies. Unfortunately, the first experiments carried out were not conclusive. To the author's opinion, the main reason for this failure is the market environment, which has completely different characteristics compared to the environments for which positive results have been reported for the DRL approach. More precisely, four major problems have been identified, without getting into too many details. First, the long-term procurement problem studied leads to a particularly sparse reward setting, which significantly complicates the learning process. Second, the dynamics of the market environment are continuously changing, with a training set that is not always representative of the test set. Third, the considerable stochasticity of the environment makes the learning process even more complicated. Last, the market environment is poorly observable, meaning that the information available to the agent is fairly limited compared to the complexity of the environment. Hopefully, further research will eventually alleviate these problems, so that successful procurement policies could be learnt on the basis of DRL algorithms. The present doctoral thesis contributes to this important research.

2.9 Conclusions

This thesis chapter introduces a novel algorithmic solution, denominated *Uniformity-based Procurement of Electricity* (UPE), advising a retailer or a large consumer of electricity for its procurement task in the forward markets. This algorithm relies on a forecasting mechanism to predict the market trend and on the concept of procurement uniformity, which quantifies the deviation from a perfectly uniform policy of reference purchasing a fixed amount of power at each time step over the entire procurement horizon. Taking advantage of the successful DL approach for the forecasting problem, the proposed solution outperforms the benchmark procurement strategies and achieves a reduction in costs of 1.7% with respect to the perfectly uniform policy realising the average price. This represents an expected yearly saving of €775,000 for an annual electricity consumption of 1 TWh. Besides these promising results, the algorithmic solution presented exhibits key advantages. Firstly, the UPE algorithm is relatively stable in its performance, with consistent results achieved throughout the years despite having to deal with various market phenomena. Secondly, the procurement policies learnt are quite robust with regard to exceptional events, such as economic crises, because of the mechanism constraining the deviation with respect to a perfectly uniform procurement policy. Lastly, the decisions advised are completely transparent and easily interpretable, which is key to improve the reliability of the algorithmic solution. To conclude, the proposed methodology could also be slightly adapted to address the sequential decision-making problem of selling electricity in the forward markets, which could benefit energy producers.

Science is organised knowledge. Wisdom is organised life.

— Emmanuel Kant

Chapter 3

An Application of Deep Reinforcement Learning to Algorithmic Trading

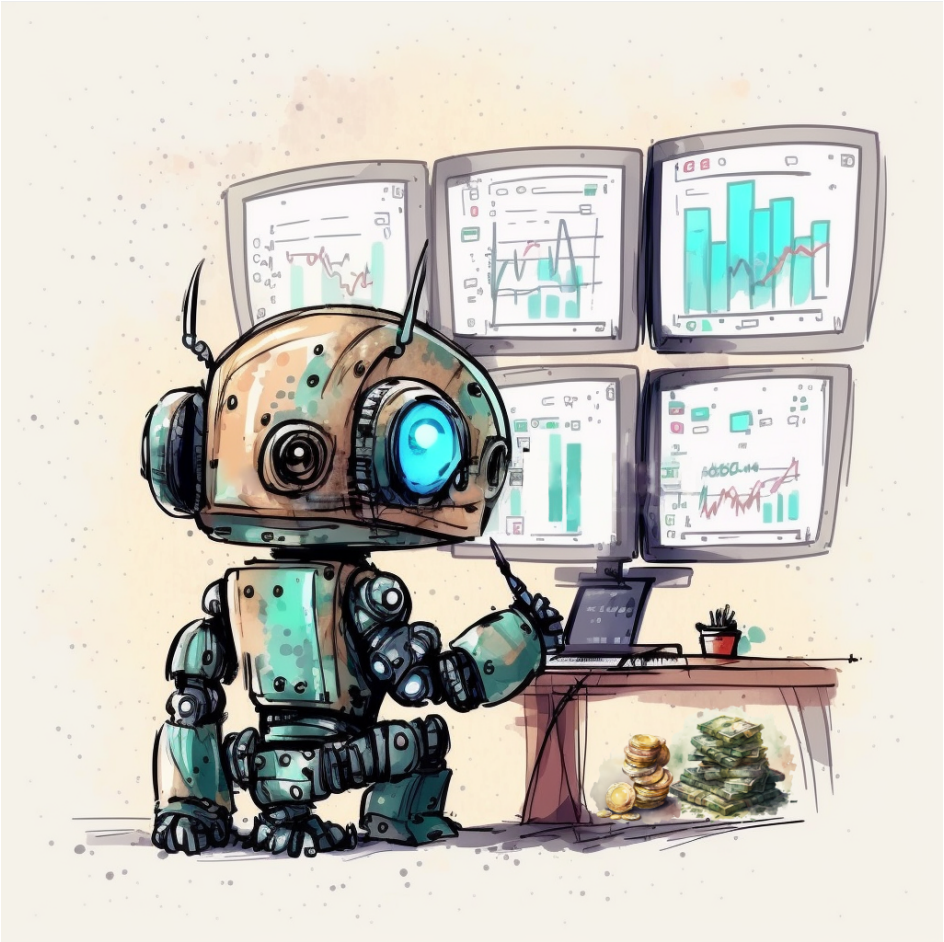


Figure 3.1: Illustration of Chapter 3 entitled *An Application of Deep Reinforcement Learning to Algorithmic Trading*, created by a generative art AI [1].

Chapter overview

This thesis chapter presents a novel approach on the basis of innovative deep reinforcement learning (DRL) techniques to deal with a challenging sequential decision-making problem related to algorithmic trading in the stock market. More precisely, the objective is to determine the optimal position, either long or short, at any point in time during the trading activity of a given stock. To achieve that goal, a trading policy taking as input the recent history of the price signal is learnt by reinforcement, so as to maximise the associated Sharpe ratio performance indicator. The resulting DRL algorithm is denominated *Trading Deep Q-Network* (TDQN), drawing inspiration from the successful DQN algorithm and adapted to the particular algorithmic trading decision-making problem studied. Both training and testing of this novel DRL algorithm are based exclusively on the generation of artificial trajectories from a limited set of stock market historical data. In order to objectively assess the performance of the trading policies learnt, this research work also introduces a new and more rigorous quantitative performance assessment methodology. The realistic experiments conducted highlight the strengths and weaknesses of the TDQN algorithm, for which promising results are reported.

This thesis chapter is primarily based on the following scientific publication [3]:

Thibaut Théate and Damien Ernst. An Application of Deep Reinforcement Learning to Algorithmic Trading. *Expert Systems with Applications*, 173:114632, 2021.

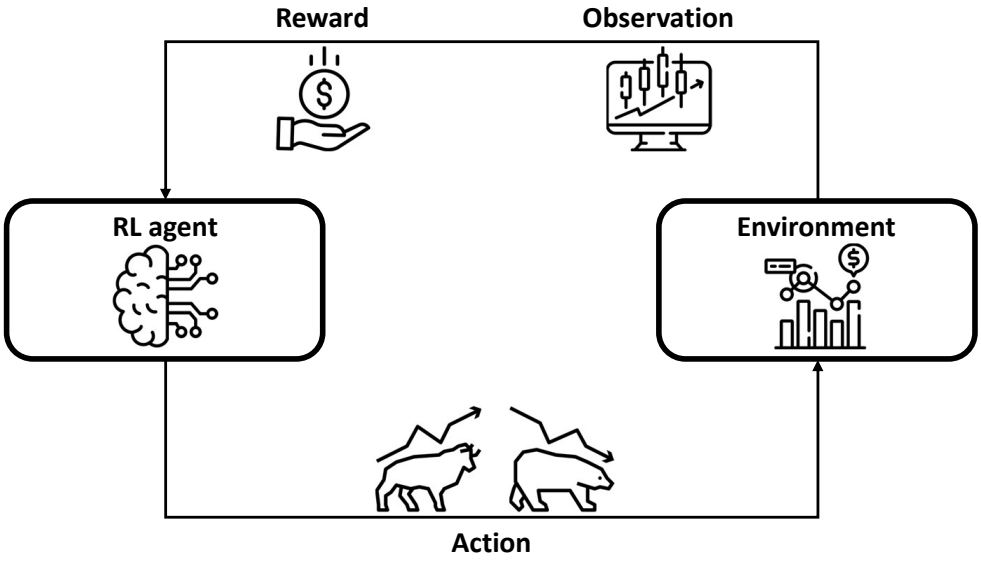


Figure 3.2: General illustration of the novel technical solution presented in this thesis chapter entitled *An Application of Deep Reinforcement Learning to Algorithmic Trading*.

3.1 Introduction

For the past few years, the interest in *artificial intelligence* (AI) has grown at a very fast pace, with numerous research papers published every year. A key element for this growing interest is related to the impressive successes of *deep learning* (DL) techniques, which are based on *deep neural networks* (DNNs). The latter may be viewed as mathematical models directly inspired by the human brain structure. This innovative approach has nowadays become the state of the art in numerous applications such as speech recognition, image classification, natural language processing or generative models. In parallel to DL, another field of research has recently gained much more attention from the research community: *deep reinforcement learning* (DRL). In short, this family of techniques is concerned with the learning process of an intelligent agent (i) interacting in a sequential manner with an unknown environment (ii) aiming to maximise its cumulative rewards and (iii) taking advantage of DL techniques to generalise the information acquired from its interactions with the environment. Without going into detail, the multiple recent successes of DRL techniques highlight their ability to solve complex sequential decision-making problems.

Nowadays, an emerging industry that is growing extremely fast is the *financial technology* industry, generally referred to by the abbreviation *FinTech*. The objective of FinTech is pretty simple: to extensively take advantage of technology in order to innovate and improve activities in finance. In the coming years, the FinTech industry is expected to revolutionise the way diverse decision-making problems related to the financial sector are addressed, including the problems related to trading, investment, portfolio management, risk management, fraud detection or financial advising. In fact, such decision-making problems are particularly complex to solve as they generally are sequential in nature and characterised by considerable stochasticity, with a partially observable and potentially adversarial environment. Especially, *algorithmic trading*, which is a key sector of the FinTech industry, presents really interesting challenges. More precisely, also called quantitative trading, algorithmic trading may be viewed as a methodology to trade using computers and a specific set of mathematical rules.

The primarily objective of this research work is to answer the following question: how to design a novel trading policy/strategy on the basis of innovative AI techniques that could compete with the popular algorithmic trading strategies that are widely adopted in practice? To achieve that goal, this thesis chapter presents and analyses a new DRL solution to tackle the algorithmic trading problem of determining the optimal trading position (long or short) at any point in time during a trading activity in the stock market. The algorithmic solution presented in this research work is inspired by the popular Deep Q-Network (DQN) algorithm, which is adapted to the particular sequential decision-making problem studied. Additionally, a quantitative performance assessment is introduced to evaluate in a rigorous and unbiased way the performance of a trading policy in the stock market. Finally, the experiments conducted are particularly instructive for the DRL approach since the trading environment at hand presents different characteristics from those that have already been successfully solved by DRL methods, mainly significant stochasticity and extremely poor observability.

3.2 Literature review

To begin this scientific literature review, two key facts have to be emphasised. Firstly, it is important to be aware that many sound research works in the field of algorithmic trading are not publicly available. As explained in article [34], due to the huge amount of money at stake, private FinTech firms are very unlikely to make their latest research results public. Secondly, it has to be acknowledged that making a fair comparison between trading strategies is a challenging task, because of the lack of a well-established framework to properly evaluate their performance. Instead, the authors generally define their own benchmark with their potential bias. Additionally, another major problem is related to the trading costs which are variously defined or even omitted in the literature.

Most of the research works in algorithmic trading concern techniques generally developed by mathematicians, economists and traders who do not exploit the recent breakthroughs in AI. Typical examples of classical algorithmic trading strategies are the *trend following* and *mean reversion* strategies, which are covered in detail in [19], [35] and [36]. The majority of works applying *machine learning* (ML) techniques in the algorithmic trading field focus on forecasting. If the financial market evolution is known in advance with a reasonable level of confidence, the optimal trading decisions can easily be computed. Following this approach, several DL techniques have already been investigated with promising results. For instance, [37] introduces a new trading strategy based predictions achieved by a DNN, and [38] uses wavelet transforms, stacked autoencoders and long short-term memory (LSTM) to improve these predictions. Alternatively, some authors have investigated RL techniques to solve the algorithmic trading decision-making problem. For instance, [39] introduces a recurrent RL algorithm for discovering new investment strategies without the need to build forecasting models, and [40] uses adaptive RL to trade in foreign exchange markets. More recently, a few research works investigated DRL techniques in a scientifically sound way to solve the algorithmic trading problem. For instance, one can first mention [41] which introduces the fuzzy recurrent DNN structure to obtain a technical-indicator-free trading system that takes advantage of fuzzy learning to reduce the time series uncertainty. One can also mention [42] which studies the application of the deep Q-learning algorithm for trading in foreign exchange markets. Finally, there exist a few interesting research works studying the application of DRL techniques to algorithmic trading in specific markets, such as in the field of energy [7].

To finish this short literature review, a sensitive problem in the scientific literature is the tendency to prioritise the communication of good results or findings, sometimes at the cost of a proper scientific approach with objective criticism. Going even further, the article [43] even states that most published research findings in certain sensitive fields are probably false. Such a concern appears to be all the more relevant in the field of financial sciences, especially when the subject directly relates to trading activities. Indeed, the paper [44] claims that many scientific publications in finance suffer from the lack of a proper scientific approach, instead getting closer to pseudo-mathematics and financial charlatanism. Aware of these concerning tendencies, the present research work intends to deliver an unbiased, rigorous scientific evaluation of the novel DRL solution proposed.

3.3 Algorithmic trading problem formalisation

3.3.1 Algorithmic trading

Also known as quantitative trading, algorithmic trading is a subfield of FinTech that may be viewed as the approach of automatically making trading decisions on the basis of a set of mathematical rules computed by a machine. This commonly accepted definition is adopted in this research work, although other definitions exist in the literature. Indeed, several authors differentiate the trading decisions (quantitative trading) from the actual trading execution (algorithmic trading). However, for the sake of generality, algorithmic trading and quantitative trading are considered synonyms in this research work, defining the entire automated trading process. Algorithmic trading has already proven to be positive to markets, the main benefit being the significant improvement in liquidity, as discussed in paper [45]. For more information about this specific field, please refer to the articles [46] and [47].

There are many different markets that are suitable to apply algorithmic trading strategies. For instance, stocks and shares are traded in the stock market. Forex trading is concerned with foreign currencies. A trader could also invest in commodity futures. The recent rise of cryptocurrencies, such as the Bitcoin, offer new interesting possibilities as well. Ideally, the solution based on DRL techniques which is developed in this research work has to be applicable to multiple markets. In this thesis chapter, the focus is set exclusively on the stock market, but the conclusions drawn are expected to stand for algorithmic trading in general. As future work, the algorithmic solution introduced could be adapted/specialised to other markets as well.

Without loss of generality, a trading activity may be considered as the management of a portfolio, which is a set of assets including diverse stocks, bonds, commodities, currencies, among others. In the scope of this research work, the portfolio is assumed to consist of one single stock together with the agent's cash. Therefore, the portfolio value v_t is composed of both the trading agent's cash value v_t^c and the share value v_t^s, which continuously evolve over time t. In such a context, buying and selling operations are simply cash and share exchanges. The trading agent interacts with the stock market through an order book, which contains the entire set of buying orders (*bids*) and selling orders (*asks*). An example of a simplified order book is depicted in Table 3.1. An order represents the willingness of a market participant to trade and is composed of a price p, a quantity q and a side s (either bid or ask). For a trade to occur, a match between bid and ask orders is required, an event which can only occur if $p_{max}^{bid} \geq p_{min}^{ask}$, with p_{max}^{bid} (p_{min}^{ask}) being the maximum (minimum) price of a bid (ask) order. Consequently, a trading agent faces a very challenging task in order to generate profit: what, when, how, at which price and which quantity to trade? This is the complex sequential decision-making problem behind algorithmic trading which is studied in this research work.

Table 3.1: Example of a simplified order book.

Side s	**Quantity** q	**Price** p
Ask	3000	107
Ask	1500	106
Ask	500	105
Bid	1000	95
Bid	2000	94
Bid	4000	93

3.3.2 Timeline discretisation

In reality, the trading decisions can be issued at any time within the business hours, making trading a continuous process. In order to study the algorithmic trading problem previously described, a discretisation operation of the continuous timeline is performed. More precisely, the trading timeline is discretised into a number of discrete trading time steps t of constant duration Δt. In the thesis chapter, for the sake of readability, the increment (decrement) operations $t+1$ $(t-1)$ are used to model the discrete transition from time step t to time step $t+\Delta t$ $(t-\Delta t)$.

The duration Δt is closely linked to the trading frequency targeted by the trading agent (very high trading frequency, intraday, daily, monthly). Such a discretisation operation inevitably imposes a constraint with respect to this trading frequency. Indeed, because the duration Δt between two time steps can not be chosen as small as possible due to technical constraints, the maximum trading frequency achievable, equal to $1/\Delta t$, is limited. In the scope of this research work, this constraint is met as the trading frequency targeted is daily, meaning that the trading agent makes a new decision once every day.

3.3.3 Trading strategy

The algorithmic trading approach is rule-based, meaning that the trading decisions are made according to a set of rules, which are embedded into a *trading strategy*. In technical terms, a trading strategy may be viewed as a programmed policy $\pi(a_t|i_t)$, either deterministic or stochastic, which outputs a trading action a_t according to the information available to the trading agent i_t at time step t. Additionally, a key characteristic of a trading strategy is its sequential aspect, as illustrated in Figure 3.3. An agent executing its trading strategy sequentially applies the following steps:

1. Update of the available market information i_t.
2. Execution of the strategy $\pi(a_t|i_t)$ to get action a_t.
3. Application of the designated trading action a_t.
4. Next time step $t \rightarrow t+1$, loop back to step 1.

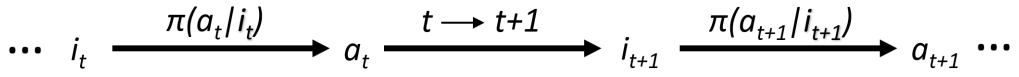


Figure 3.3: Illustration of the sequential execution of a trading strategy.

3.3.4 Reinforcement learning problem formalisation

As illustrated in Figure 3.4 hereafter, *reinforcement learning* (RL) is concerned with the sequential interaction of an agent with its environment. At each time step t, the RL agent firstly observes the environment of internal state s_t, and retrieves an observation o_t. Then, it executes the action a_t resulting from its policy $\pi(a_t|h_t)$, with h_t being the RL agent history, and receives a reward r_t as a consequence of its action. In this RL context, the agent history can be mathematically, expressed as $h_t = \{(o_\tau, a_\tau, r_\tau)|\tau = 0, 1, ..., t\}$. Reinforcement learning techniques are concerned with the design of policies π maximising an optimality criterion, which directly depends on the immediate rewards r_t observed over a certain time horizon. The most popular optimality criterion is the expected discounted sum of rewards over an infinite time horizon, also named return and which is denoted R. Formally, the resulting optimal policy π^* is expressed as the following:

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, \mathbb{E}[R|\pi] \ , \tag{3.1}$$

$$R = \sum_{t=0}^{\infty} \gamma^t r_t \ . \tag{3.2}$$

The parameter $\gamma \in [0, 1]$ is the discount factor, which determines the relative importance of future rewards. For instance, if $\gamma = 0$, the RL agent is said to be myopic as it only considers the current reward and totally discards the future rewards. When the discount factor increases, the RL agent tends to become more long-term oriented. In the extreme case with $\gamma = 1$, the RL agent considers each reward equally. This important parameter has to be carefully tuned according to the desired behaviour.

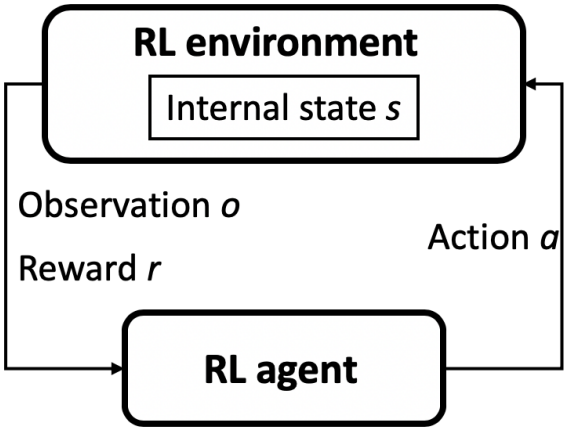


Figure 3.4: Illustration of the reinforcement learning approach.

Algorithmic trading: observation space

In the scope of the algorithmic trading problem studied, the environment is assumed to be the entire trading world gravitating around the RL agent. In fact, this market environment can be viewed as an abstraction including the trading mechanisms together with every single piece of information that could potentially have an effect on the trading activity of the agent. A major challenge of the algorithmic trading problem is the extremely poor observability of this market environment. Indeed, a considerable amount of information is completely hidden to the trading agent, ranging from some companies' confidential information to the other market participants' strategies. Therefore, the information available to the trading agent is extremely limited compared to the complexity of the environment. From the perspective of the RL agent, this makes the market environment particularly stochastic. Additionally, the information available may take various forms, being both quantitative and qualitative. Correctly processing such information and re-expressing it using relevant quantitative figures while minimising the subjective bias is capital. Finally, the information retrieved at each time step has to be taken into consideration sequentially rather than individually.

At each trading time step t, the RL agent observes the stock market whose internal state is $s_t \in \mathcal{S}$. The limited information collected by the agent on this complex market environment is denoted $o_t \in \mathcal{O}$. Ideally, this observation space $\mathcal{O}$ has to encompass all the information capable of influencing the market prices. As previously explained, because of the sequential nature of the algorithmic trading problem, an observation o_t has to be considered as a sequence of both the information gathered during the previous τ time steps (history) and the newly available information at time step t. In this research work, the RL agent's observations can be mathematically expressed as the following:

$$o_t = \{S(t'),\ D(t'),\ T(t'),\ I(t'),\ M(t'),\ N(t'),\ E(t')\}_{t'=t-\tau}^{t}\ , \tag{3.3}$$

where:

- $S(t)$ represents the state information of the RL agent at time step t, such as the current trading position, the number of shares owned or the available cash.
- $D(t)$ is the information gathered by the agent at time step t concerning the OHLCV (Open-High-Low-Close-Volume) data characterising the stock market. More precisely, $D(t)$ can be formally expressed as follows:

$$D(t) = \{p_t^O,\ p_t^H,\ p_t^L,\ p_t^C,\ V_t\}, \tag{3.4}$$

  where:

  - p_t^O is the stock market price at the opening of the time period $[t - \Delta t,\ t[$,
  - p_t^H is the highest stock market price over the time period $[t - \Delta t,\ t[$,
  - p_t^L is the lowest stock market price over the time period $[t - \Delta t,\ t[$,
  - p_t^C is the stock market price at the closing of the time period $[t - \Delta t,\ t[$,
  - V_t is the total volume of shares exchanged over the time period $[t - \Delta t,\ t[$.

- $T(t)$ is the information regarding the trading time step t itself (date, weekday, time).
- $I(t)$ is the information at the disposal of the agent regarding multiple technical indicators about the stock market targeted at time step t. There exist many technical indicators providing extra insights about diverse financial phenomena, such as the moving average convergence divergence (MACD), relative strength index (RSI) or average directional index (ADX), to only cite the most popular ones.
- $M(t)$ gathers the macroeconomic information at the disposal of the agent at time step t. There are many interesting macroeconomic indicators which could potentially be useful to forecast the evolution of markets, such as the interest rate or the exchange rate.
- $N(t)$ represents the news information gathered by the agent at time step t. These key data can be retrieved from various sources such as the social media (Facebook, Twitter, LinkedIn), some specialised websites, or the newspapers. Complex sentiment analysis models have to be designed in order to extract meaningful quantitative figures (volume, sentiment polarity and subjectivity) from the news. The benefits of such information have already been demonstrated by several authors, see e.g. papers [48], [49] and [50].
- $E(t)$ is any extra useful information at the disposal of the RL agent at time step t, including among others the strategies of other market participants, some companies' confidential information, the behaviour of related stock markets, advice from experts, but also rumours.

In the scope of this research work, it is assumed that the only information taken into consideration by the RL agent is the classical OHLCV data $D(t)$ together with the state information $S(t)$. More precisely, the reduced observation space $\mathcal{O}$ encompasses the current trading position together with a series of the previous $\tau + 1$ daily open-high-low-close prices and daily traded volume. Following such an assumption, the reduced RL observation o_t at time step t can be mathematically expressed as the following:

$$o_t = \left\{ \{ p_{t'}^O,\ p_{t'}^H,\ p_{t'}^L,\ p_{t'}^C,\ V_{t'} \}_{t'=t-\tau}^{t},\ P_t \right\}, \tag{3.5}$$

with P_t being the trading position of the RL agent at time step t, either *long* or *short*.

Algorithmic trading: action space

At each time step t, the RL agent executes a trading action $a_t \in \mathcal{A}$ resulting from its decision-making policy $\pi(a_t|o_t)$. In fact, the trading agent has to answer several questions: whether, how and how much to trade? This decision can be modelled by the quantity of shares purchased by the trading agent at time step t, represented by $Q_t \in \mathbb{Z}$. Consequently, the RL action a_t at time step t can be formally expressed as the following:

$$a_t = Q_t\ . \tag{3.6}$$

Three cases have to be differentiated according to the value of Q_t:

- $Q_t > 0$: The agent *buys* shares on the stock market, by posting new *bid* orders.
- $Q_t < 0$: The agent *sells* shares on the stock market, by posting new *ask* orders.
- $Q_t = 0$: The agent *holds*, not buying nor selling any shares on the stock market.

As previously hinted, the actions occurring in practice in the scope of a trading activity are the orders posted in the order book. Therefore, the RL agent is assumed to communicate with an external module responsible for the synthesis of these real actions according to the value of Q_t: the *trading execution system*. Nevertheless, this other decision-making problem is out of the scope of this research work and is not discussed further.

Naturally, the trading actions have an impact on the two components of the portfolio value, namely the cash and share values (v_t^c and v_t^s). In this research work, it is assumed that the trading action a_t at time step t occurs just before market closure at a price $p_t \simeq p_t^C$. Following this assumption, the portfolio value can updated as the following:

$$v_{t+1}^c = v_t^c - Q_t \; p_t \; , \tag{3.7}$$

$$v_{t+1}^s = \underbrace{(n_t + Q_t)}_{n_{t+1}} \; p_{t+1} \; , \tag{3.8}$$

with $n_t \in \mathbb{Z}$ being the number of shares owned by the agent at time step t. Interestingly, this research work allows negative values for that quantity. Despite being surprising at first glance, a negative number of shares simply corresponds to shares borrowed and sold, with the obligation to repay the lender in shares in the future. Such a mechanism is particularly important to take into consideration as it introduces new possibilities for the trading agent.

Two capital constraints are assumed regarding the quantity of shares traded Q_t. Firstly, contrarily to the share value v_t^s which can be both positive or negative, the cash value v_t^c has to remain positive for every trading time steps t. This constraint imposes an upper bound on the number of shares that the trading agent is capable of purchasing, which can be easily derived from Equation (3.7). Secondly, there exists a risk associated with the impossibility to repay the share lender if the agent suffers significant losses. To prevent such a situation from happening, the cash value v_t^c is constrained to be sufficiently large when a negative number of shares is owned, in order to be able to get back to a neutral position ($n_t = 0$). A maximum relative change in price, expressed in % and denoted $\epsilon \in \mathbb{R}^+$, is assumed by the agent prior to the trading activity. This parameter corresponds to the maximum daily evolution of the market price supposed by the agent over the entire trading horizon, so that the trading agent should always be capable of paying back the share lender as long as the price variation remains below this value. Consequently, the constraints acting upon the RL actions at time step t can be mathematically expressed as follows:

$$v_{t+1}^c \geq 0 \; , \tag{3.9}$$

$$v_{t+1}^c \geq -n_{t+1} \; p_t \; (1 + \epsilon) \; , \tag{3.10}$$

with the following condition assumed to be satisfied:

$$\left| \frac{p_{t+1} - p_t}{p_t} \right| \leq \epsilon \ . \tag{3.11}$$

Actually, the modelling represented by Equation (3.7) is inaccurate and will inevitably lead to unrealistic results. Indeed, whenever simulating trading activities, the trading costs have to be carefully taken into consideration. Such an omission can potentially be misleading because a trading strategy, which is highly profitable in simulations, may generate large losses in real trading situations caused by these trading costs, especially when the trading frequency is high. The trading costs can be subdivided into two categories. On the one hand, the explicit costs are induced by transaction costs and taxes. On the other hand, there are implicit costs, also called slippage costs, which are composed of three main components and are associated to some of the dynamics of the market environment:

- **Spread costs.** These costs are related to the difference between the minimum ask price p_{min}^{ask} and the maximum bid price p_{max}^{bid}, called the *spread*. Because the complete state of the order book is generally too complex to efficiently process or even not available, the trading decisions are mostly based on the middle price $p^{mid} = (p_{max}^{bid} + p_{min}^{ask})/2$. However, a buying (selling) trade issued at p^{mid} inevitably occurs at a price $p \geq p_{min}^{ask}$ ($p \leq p_{max}^{bid}$). Such costs are all the more significant that the stock market liquidity is low compared to the quantity of shares traded.
- **Market impact costs.** These costs are induced by the impact of the trader's actions on the market. Each trade, whether it consists of buying or selling orders, is potentially capable of influencing the market price. This phenomenon is all the more pronounced that the stock market liquidity is low with respect to the quantity of shares traded.
- **Timing costs.** These costs are caused by the time required for a trade to physically happen once the trading decision is made, since the market price is continuously evolving. The first cause is the inevitable latency that delays the posting of the orders in the market order book. The second cause is the intentional delays that may potentially be generated by the trading execution system. For instance, a large trade could be split into multiple smaller trades spread over time in order to limit the market impact costs.

As previously explained, an accurate modelling of the trading costs is required to faithfully reproduce the dynamics of a real trading environment. While explicit costs are relatively easy to take into consideration, the valid modelling of slippage costs is a truly complex task. In this research work, the integration of both costs into the RL environment is performed through a heuristic. When a trade is performed, a certain amount of capital equivalent to a percentage $C \in \mathbb{R}^+$ of the amount of money invested is lost. This parameter was realistically chosen equal to 0.1% in the forthcoming simulations. Practically, these trading costs are directly withdrawn from the trading agent's cash, leading to the following equation which is the re-expression of Equation 3.7 with a corrective term modelling the trading costs:

$$v_{t+1}^c = v_t^c - Q_t \ p_t - \underbrace{C \ |Q_t| \ p_t}_{\text{Trading costs}} \ . \tag{3.12}$$

Moreover, the trading costs have to be properly included into the constraint expressed in Equation (3.10). Indeed, the cash value v_t^c has to be sufficiently large to get back to a neutral position ($n_t = 0$) when the maximum price variation ϵ occurs, with the trading costs being included. Therefore, Equation (3.10) can be re-expressed as follows:

$$v_{t+1}^c \geq -n_{t+1}\ p_t\ (1+\epsilon)(1+C). \tag{3.13}$$

Eventually, the RL action space $\mathcal{A}$ can be defined as the discrete set of acceptable values for the quantity of shares traded Q_t, with regard to the two constrained previously explained. Formally proven hereafter, the RL action space $\mathcal{A}$ can be mathematically expressed as follows:

$$\mathcal{A} = \{Q_t \in \mathbb{Z} \cap [\underline{Q_t},\ \overline{Q_t}]\}, \tag{3.14}$$

where:

- $\overline{Q_t} = \frac{v_t^c}{p_t(1+C)}$,
- $\underline{Q_t} = \begin{cases} \frac{\Delta_t}{p_t\epsilon(1+C)} & \text{if } \Delta_t \geq 0, \\ \frac{\Delta_t}{p_t(2C+\epsilon(1+C))} & \text{if } \Delta_t < 0, \end{cases}$
  with $\Delta_t = -v_t^c - n_t\ p_t\ (1+\epsilon)(1+C)$.

Theorem 1. *On the basis of the algorithmic trading context previously described together with the assumptions made, the RL action space $\mathcal{A}$ admits an upper bound $\overline{Q_t}$ such that:*

$$\overline{Q_t} = \frac{v_t^c}{p_t(1+C)}\ .$$

Proof. The upper bound of the RL action space $\mathcal{A}$ is derived from the assumption that the cash value v_t^c has to always remain positive, as formally expressed in Equation (3.9). Making the hypothesis that $v_t^c \geq 0$, the number of shares Q_t traded by the RL agent at time step t has to be such that the constraint $v_{t+1}^c \geq 0$ is respected. Introducing the former constraint into Equation (3.12) describing the update of the cash value, the following inequality is obtained:

$$v_t^c - Q_t\ p_t - C\ |Q_t|\ p_t \geq 0\ .$$

Two different cases arise depending on the value of Q_t:

Selling shares ($Q_t < 0$):
$v_t^c - Q_t\ p_t + C\ Q_t\ p_t \geq 0$
$\Leftrightarrow Q_t \leq \frac{v_t^c}{p_t(1-C)}$.
The right hand side of this inequality is always positive because of the hypothesis that $v_t^c \geq 0$. Since Q_t is strictly negative in this case (selling of shares), the condition is always satisfied.

Buying shares ($Q_t \geq 0$):
$v_t^c - Q_t\ p_t - C\ Q_t\ p_t \geq 0$
$\Leftrightarrow Q_t \leq \frac{v_t^c}{p_t(1+C)}$.
This condition represents the (positive) upper bound of the RL action space $\mathcal{A}$. □

Theorem 2. *On the basis of the algorithmic trading context previously described together with the assumptions made, the RL action space $\mathcal{A}$ admits a lower bound $\underline{Q_t}$ such that:*

$$\underline{Q_t} = \begin{cases} \frac{\Delta_t}{p_t\epsilon(1+C)} & \text{if } \Delta_t \geq 0, \\ \frac{\Delta_t}{p_t(2C+\epsilon(1+C))} & \text{if } \Delta_t < 0, \end{cases}$$

with $\Delta_t = -v_t^c - n_t \ p_t \ (1+\epsilon)(1+C)$.

Proof. The lower bound of the RL action space $\mathcal{A}$ is derived from the assumption that the cash value v_t^c has to always be sufficient to get back to a neutral position ($n_t = 0$), as formally expressed in Equation (3.13). Making the hypothesis that this constraint is satisfied at time step t, the number of shares Q_t traded by the RL agent has to be such that this condition remains true at the next time step $t+1$. Introducing this constraint into Equation (3.12), the following inequality is obtained:

$$v_t^c - Q_t \ p_t - C \ |Q_t| \ p_t \geq -(n_t + Q_t) \ p_t \ (1+C)(1+\epsilon).$$

Two different cases arise depending on the value of Q_t:

Buying shares ($Q_t \geq 0$):
$v_t^c - Q_t \ p_t - C \ Q_t \ p_t \geq -(n_t + Q_t) \ p_t \ (1+C)(1+\epsilon)$
$\Leftrightarrow Q_t \geq \frac{-v_t^c - n_t p_t (1+C)(1+\epsilon)}{p_t \epsilon (1+C)}$
This condition represents the first lower bound of the RL action space $\mathcal{A}$.

Selling shares ($Q_t < 0$):
$v_t^c - Q_t \ p_t + C \ Q_t \ p_t \geq -(n_t + Q_t) \ p_t \ (1+C)(1+\epsilon)$
$\Leftrightarrow Q_t \geq \frac{-v_t^c - n_t p_t (1+C)(1+\epsilon)}{p_t (2C + \epsilon(1+C))}$
This condition represents the second lower bound of the RL action space $\mathcal{A}$.

Both lower bounds previously derived have the same numerator, which is denoted Δ_t. In fact, this quantity represents the difference between the maximum assumed cost to get back to a neutral position at the next time step $t+1$ and the current cash value of the agent v_t^c. In other words, it indicates whether or not the trading agent is able to pay its debt in shares in the worst assumed case, if no shares are traded at the current time step ($Q_t = 0$). Once again, two cases arise depending on the sign of the quantity Δ_t:

Case of $\Delta_t < 0$: The trading agent has no problem paying its debt in shares in the situation previously described. Indeed, this statement is always true when the agent owns a positive number of shares ($n_t \geq 0$). Moreover, this is also always true when the agent owns a negative number of shares ($n_t < 0$) and when the price decreases ($p_t < p_{t-1}$), on the basis of the hypothesis that Equation (3.13) was satisfied at time step t. In this case, the most constraining lower bound of the two is the following:

$$\underline{Q_t} = \frac{\Delta_t}{p_t(2C + \epsilon(1+C))} \ .$$

Case of $\Delta_t \geq 0$: The trading agent may potentially have problem paying its debt in shares in the situation previously described. Following a similar reasoning as for the previous case, the most constraining lower bound of the two is the following:

$$\underline{Q_t} = \frac{\Delta_t}{p_t \epsilon (1 + C)} \ .$$

□

In the scope of this research work, the RL action space $\mathcal{A}$ is slightly reduced in order to lower the complexity of the algorithmic trading problem. More precisely, the reduced action space becomes discrete and offers two different possibilities to the agent. At each time step t, the RL agent can either choose to adopt a *long* trading position or a *short* trading position for the next time step, denoted P_{t+1}. A long trading position consists in purchasing shares now in order to sell them back in the future, hopefully at a higher price. On the contrary, a short trading position consists in borrowing some shares to be sold now with the obligation to buy them back in the future, ideally at a lower price. Eventually, the two actions available can be mathematically expressed as the following:

$$a_t = Q_t \in \{Q_t^{Long},\ Q_t^{Short}\}. \tag{3.15}$$

The first RL action Q_t^{Long} is designed to maximise the number of shares owned by the trading agent, by converting as much cash value v_t^c as possible into share value v_t^s. Formally, it can be mathematically expressed as follows:

$$Q_t^{Long} = \begin{cases} \left\lfloor \frac{v_t^c}{p_t(1+C)} \right\rfloor & \text{if } a_{t-1} \neq Q_{t-1}^{Long}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.16}$$

The action Q_t^{Long} is always valid since it is included into the original action space $\mathcal{A}$ defined in Equation 3.14. Executing this action results in the trading agent owning a number of shares $N_t^{Long} = n_t + Q_t^{Long}$. On the contrary, the second action, denoted Q_t^{Short}, converts share value v_t^s into cash value v_t^c, in such a way that the trading agent owns a number of shares equal to $-N_t^{Long}$. This operation can be mathematically expressed as the following:

$$\widehat{Q_t^{Short}} = \begin{cases} -2n_t - \left\lfloor \frac{v_t^c}{p_t(1+C)} \right\rfloor & \text{if } a_{t-1} \neq Q_{t-1}^{Short}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.17}$$

Nevertheless, the action $\widehat{Q_t^{Short}}$ may potentially violate the lower bound $\underline{Q_t}$ of the action space $\mathcal{A}$ when the price significantly increases over time. Consequently, the second action Q_t^{Short} is formally expressed as follows:

$$Q_t^{Short} = \max\left\{\widehat{Q_t^{Short}},\ \underline{Q_t}\right\}. \tag{3.18}$$

Algorithmic trading: reward design

In the scope of the algorithmic trading problem studied, a relevant choice for the RL rewards would be the daily returns achieved by the trading policy, for three reasons. Firstly, it makes sense intuitively to seek the highest returns which are an evidence of a profitable trading strategy. Secondly, the daily return presents the advantage of being independent of the number of shares owned by the agent. Lastly, considering the daily returns as rewards allows to avoid a sparse reward setup, which generally leads to a much more complicated learning process. Therefore, the RL rewards are mathematically defined as the following:

$$r_t = \frac{v_{t+1} - v_t}{v_t} \ . \tag{3.19}$$

Algorithmic trading: RL objective

Exhaustively assessing the performance of a trading strategy is a challenging task, because of the numerous quantitative and qualitative factors to ideally take into consideration. In fact, a successful trading policy is not solely expected to generate convenient profit, but also to efficiently mitigate the risk associated with the trading activity. Naturally, the balance (generally trade-off) between these two objectives may significantly vary depending on the trader's profile together with its willingness to take extra risks in order to potentially achieve a higher profit. Consequently, despite being intuitively appropriate, maximising the profit generated by a trading strategy is a necessary but not sufficient objective. Instead, the core objective of a trading strategy is selected to be the maximisation of the *Sharpe ratio*, a performance indicator widely used in the field of finance and algorithmic trading. Its strength lies in the simultaneous consideration of both the generated profit and the risk associated with the trading activity. Mathematically, the Sharpe ratio S_r is expressed as the following:

$$S_r = \frac{\mathbb{E}[R_s - R_f]}{\sigma_r} = \frac{\mathbb{E}[R_s - R_f]}{\sqrt{\text{var}[R_s - R_f]}} \simeq \frac{\mathbb{E}[R_s]}{\sqrt{\text{var}[R_s]}} \ , \tag{3.20}$$

where:

- R_s is the return achieved by the trading strategy over a certain period of time, modelling the profitability of the trading policy π learnt,
- R_f is the risk-free return over that same time period, meaning the expected return from a perfectly safe investment (considered negligible in this research work),
- σ_r is the standard deviation of the excess return $R_s - R_f$ achieved by the trading strategy, modelling the riskiness of the trading activity.

In practice, the Sharpe ratio S_r is computed as follows. Firstly, the daily returns achieved by the trading strategy are computed using the formula $\rho_t = (v_t - v_{t-1})/v_{t-1}$. Secondly, the ratio between the mean and standard deviation of these daily returns is evaluated. Lastly, the annualised Sharpe ratio is derived by multiplying the former result with the square root of the number of trading days within a year (252).

Although the core objective targeted is the maximisation of the Sharpe ratio quantitative indicator, the DRL approach adopted in this research work actually maximises the expected discounted sum of daily returns (rewards) over an infinite time horizon. In fact, the former optimisation criterion, despite being different, may be regarded as a relaxation of the Sharpe ratio criterion. An interesting future research direction could be to further narrow the gap between these two objectives.

Moreover, a successful trading strategy has to ideally be capable of achieving convenient performance for different markets presenting diverse patterns. Typically, the perfect trading strategy is expected to achieve great results for both bullish and bearish markets together with various levels of volatility, referring to an upward and downward trend in the price evolution, respectively. Therefore, the ultimate objective of this research work is the design of a novel trading strategy on the basis of innovative DRL techniques in order to maximise the Sharpe ratio on average for various market patterns.

3.4 Deep reinforcement learning algorithm

3.4.1 Deep Q-Network algorithm

The Deep Q-Network algorithm, generally referred to as DQN, is a prominent DRL algorithm that may be viewed as the successor of the popular Q-learning algorithm [51]. This approach is said to be *model-free*, meaning that a complete model of the environment is not required, some trajectories alone being sufficient. Belonging to the *Q-learning* family of methods, this algorithm is based on the learning of an approximation of the state-action value function, denoted Q, via the Bellman equation [52]. The innovation brought by the DQN algorithm resides in the modelling of the function Q with a DNN, whose parameters θ have to be learnt. This optimisation is primarily performed *off-policy*, meaning that each update is based on previous experiences $e_t = (s_t, a_t, r_t, s_{t+1})$ collected at any point during training.

The DQN algorithm has originally been designed as a solution to learn playing the Atari-57 benchmark [53], for which it achieved promising results [54, 55]. Since then, a great scientific literature has appeared around this algorithm, with both applications and improvements. Typically, one can mention the numerous enhancements from the Rainbow algorithm [56]: multi-step learning [57], double Q-learning [58], prioritised experience replay [59], duelling architecture [60], distributional RL [61] and noisy networks [62]. The interested reader can also refer to the following resources for more information about DL [25, 63, 22] and RL [64, 65, 66, 67, 68] in general.

3.4.2 Generation of artificial trajectories

In the scope of the algorithmic trading problem studied, a model of the environment $\mathcal{E}$ is obviously not available. Therefore, the training of the DRL algorithm will be based entirely on the generation of artificial trajectories from a limited set historical daily OHLCV data from the stock market. Formally, a trajectory τ is defined as a sequence of observations $o_t \in \mathcal{O}$, actions $a_t \in \mathcal{A}$ and rewards r_t for a number T of time steps t:

$$\tau = \Big(\{o_0, a_0, r_0\}, \{o_1, a_1, r_1\}, ..., \{o_{T-1}, a_{T-1}, r_{T-1}\}\Big).$$

Initially, the single trajectory τ which is available corresponds to the historical behaviour of the stock market, without any interaction from the RL agent. Consequently, this original trajectory is composed of the historical prices and volumes together with some *long* actions executed by the RL agent without any money to invest ($v_0^c = 0$), to model the absence of a trading activity. In order to generate a dataset to learn from, new fictive trajectories are artificially created from this unique real historical trajectory to simulate interactions between the RL agent and its market environment $\mathcal{E}$. For this approach to be scientifically acceptable and lead to realistic simulations, the stock market is assumed to be unaffected by the actions performed by the trading agent. This assumption generally holds when the number of shares manipulated by the trading agent is low with respect to the liquidity of the stock market. Finally, the artificial trajectories generated are simply composed of the sequence of historical observations associated with various sequences of trading actions from the RL agent.

Additionally, a technique is employed to slightly improve the exploration of the RL agent. It relies on the particularly low cardinality of the reduced action space, with only two actions available (long and short). At each time step t, the selected action a_t is normally performed within the market environment $\mathcal{E}$, while the opposite action a_t^- is executed within a copy of this environment, denoted $\mathcal{E}^-$. Although this approach does not replace classical exploration techniques nor solve the challenging exploration/exploitation trade-off, it allows the RL agent to continuously explore at a limited computational cost.

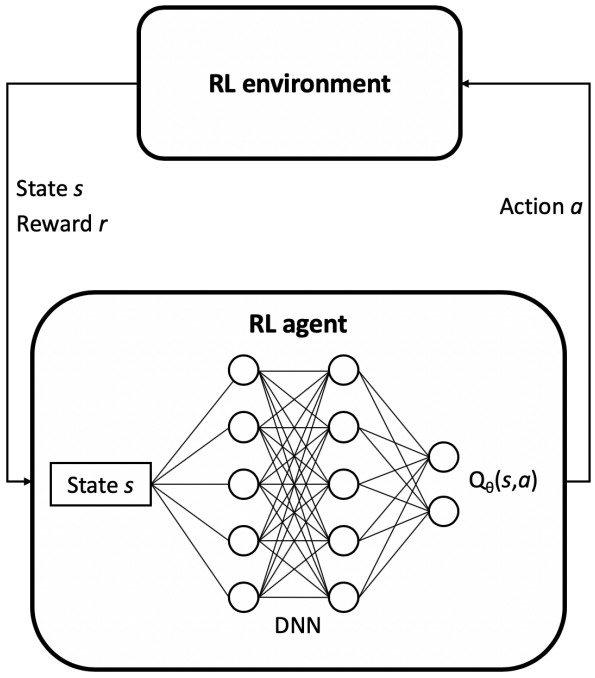


Figure 3.5: Simplified illustration of the DQN algorithm.

3.4.3 TDQN algorithm

This section presents the *Trading Deep Q-Network* (TDQN) algorithm, which is designed to learn novel algorithmic trading strategies on the basis of the DRL approach. As previously explained, the DQN algorithm is selected as a starting point, with various adaptations and improvements taken into consideration to better fit the algorithmic trading decision-making problem studied. The diverse modifications, which have been validated empirically based on the numerous simulations performed, are summarised hereafter:

- **DNN architecture.** The first important difference with respect to the original DQN algorithm is the architecture of the DNN modelling the value function $Q^\pi(s,a)$. Because of the different nature of the inputs (time-series instead of images), the *Convolutional Neural Network* (CNN) has been replaced by a classical *Feedforward Neural Network* (FNN) with some *Leaky Rectified Linear Unit* (Leaky ReLU) activation functions.
- **Xavier initialisation.** While the original DQN algorithm simply initialises the DNN weights θ randomly, the *Xavier initialisation* is implemented to improve the convergence of the DRL algorithm. The idea is to setup the initial weights in such a way that the variance of the gradients remains constant across the layers of the DNN.
- **Double DQN.** The DQN algorithm suffers from substantial overestimations, with this overoptimism harming the algorithm performance. In order to reduce the impact of this undesired phenomenon, the double DQN algorithm [58] decomposes the max operation in the target into both action selection and action evaluation.
- **ADAM optimiser.** The original DQN algorithm implements the RMSProp optimiser. However, the ADAM optimiser introduced in paper [24], has been observed to improve both the training stability and the convergence speed of the TDQN algorithm.
- **Gradient clipping.** The gradient clipping technique is implemented in the TDQN algorithm to solve the gradient exploding problem which induces significant instabilities during the training process.
- **Huber loss.** While the original DQN algorithm implements a *Mean Squared Error* (MSE) loss, the *Huber* loss experimentally improves the stability of the training phase. Such an observation may be explained by the significant penalisation of large errors with the MSE, which is generally desired but has a negative side-effect for the DQN algorithm. In this case, the DNN is supposed to output values that depend on its own input. Therefore, this DNN should not radically change in a single training update because this would also lead to an excessive change in the target as well. Ideally, the update of the DNN has to be performed in a slower and more stable manner. On the other hand, the *Mean Absolute Error* (MAE) is much more robust to the presence of outliers, but has the drawback of not being differentiable at 0. A good trade-off between the former two losses is the Huber loss H, which is formally defined as follows:

$$H(x) = \begin{cases} \frac{1}{2}x^2 & \text{if } |x| \leq 1, \\ |x| - \frac{1}{2} & \text{otherwise.} \end{cases} \tag{3.21}$$

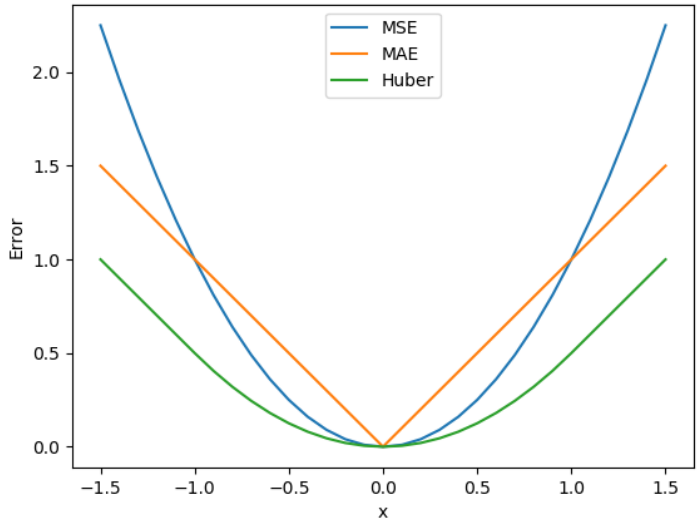


Figure 3.6: Illustration and comparison of the MSE, MAE and Huber losses.

- **Batch normalisation layers.** This DL technique, introduced by [69], consists in normalising the input layer by adjusting and scaling the activation functions. In practice, this method provides many benefits including a faster and more robust training phase together with an improved generalisation (countering overfitting).
- **Regularisation techniques.** In the preliminary experiments carried out, a very strong tendency to overfit was observed. For this reason, three regularisation techniques have been implemented: *Dropout*, *L2 Regularisation* and *Early Stopping*.
- **Preprocessing and normalisation.** Before getting manipulated by the DRL algorithm, the observations o_t go through both preprocessing and normalisation operations. Firstly, because the high-frequency noise present in the trading data has been observed to favour overfitting, a low-pass filtering operation is performed. Nevertheless, this comes at a cost since this operation modifies or even destroys some potentially useful trading patterns and introduces a non-negligible lag. Secondly, the market data are transformed in order to convey more meaningful information about market movements. Typically, the daily evolution of prices is taken into consideration instead of the raw prices. Lastly, the resulting dataset is properly normalised.
- **Data augmentation techniques.** As previously explained, a key challenge of the algorithmic trading problem studied is the limited amount of available data. In order to alleviate this critical problem, several data augmentation techniques are implemented: signal shifting, signal filtering and artificial noise addition. The idea is to generate new fictive trading data that are slightly different while keeping the underlying financial market phenomena untouched.

Eventually, for the sake of clarity and completeness, the complete TDQN algorithm is thoroughly depicted hereafter in Algorithm 2.

Algorithm 2 Trading Deep Q-Network algorithm

Initialise the experience replay memory M of capacity C.
Initialise the main DNN weights θ (Xavier initialisation).
Initialise the target DNN weights $\theta^- = \theta$.
Apply the data augmentation techniques.
Apply the preprocessing and normalisation operations.
for episode = 0 **to** N **do**
  for $t = 0$ **to** T, or until episode termination **do**
    Acquire the observation o_t from the environment $\mathcal{E}$.
    Make a virtual copy of the environment $\mathcal{E}^- = \mathcal{E}$.
    With probability ϵ, select a random action a_t from $\mathcal{A}$.
    Otherwise, select $a_t = \text{argmax}_{a \in \mathcal{A}} Q(o_t, a; \theta)$.
    Determine the (single) opposite action $a_t^- \in \mathcal{A} \setminus \{a_t\}$.
    Interact with the environment $\mathcal{E}$ with action a_t to get the next observation o_{t+1} and reward r_t.
    Interact with the environment $\mathcal{E}^-$ with action a_t^- to get the next observation o_{t+1}^- and reward r_t^-.
    Store both experiences $e_t = (o_t, a_t, r_t, o_{t+1})$ and $e_t^- = (o_t, a_t^-, r_t^-, o_{t+1}^-)$ in M.
    if $t \% T' = 0$ **then**
      Randomly sample from M a minibatch of N_e experiences $e_i = (o_i, a_i, r_i, o_{i+1})$.
      Set $y_i = \begin{cases} r_i & \text{if the observation } o_{i+1} \text{ is terminal,} \\ r_i + \gamma\ Q(o_{i+1}, \text{argmax}_{a \in \mathcal{A}} Q(o_{i+1}, a; \theta); \theta^-) & \text{otherwise.} \end{cases}$
      Compute the loss $\mathcal{L}(\theta) = H(y_i,\ Q(o_i, a_i; \theta))$.
      Clip the resulting gradient in the range [0, 1].
      Update the main DNN parameters θ using the ADAM optimiser.
    end if
    Update the target DNN parameters $\theta^- = \theta$ every N^- steps.
    Anneal the ϵ-Greedy exploration parameter ϵ.
  end for
end for

3.5 Performance assessment methodology

3.5.1 Benchmark stock markets

This section presents a novel quantitative performance assessment methodology to rigorously evaluate the soundness of algorithmic trading strategies, whether conventional or based on innovative AI techniques. This contribution is capital for producing meaningful and reliable results, especially in the financial field. As previously explained, there have been serious weaknesses concerning the evaluation protocols in algorithmic trading. In the literature, the performance of a trading strategy is commonly assessed on a single instrument over a certain period of time, either from the stock exchange or other financial markets. Nevertheless, such a methodology does not lead to reliable analyses, since the trading data could potentially have been specifically selected to artificially improve the performance of a trading strategy, which may not be as profitable in a diversified setup more representative of reality. In order to alleviate such a problematic bias, the performance of an algorithmic trading strategy has to ideally be assessed across multiple instruments characterised by diverse market patterns.

On the basis of this observation, this research work introduces a new benchmark composed of 30 stocks presenting various characteristics: different sectors and regions, both bullish and bearish markets, diverse levels of volatility but also liquidity, among others. This particular benchmark is depicted hereafter in Table 3.2. In order to avoid any confusion, the official reference for each stock (ticker) is specified in parentheses. Therefore, the ultimate objective is to design an algorithmic trading strategy achieving the best average performance for this benchmark. However, to avoid any ambiguities about the training and evaluation protocols, it should be specified that a new trading strategy can be trained for each stock considered individually, while keeping the hyperparameters of the algorithm unchanged.

Table 3.2: Benchmark adopted within the performance assessment methodology.

Sector	**Region**		
	American	*European*	*Asian*
Trading index	Dow Jones (DIA) S&P 500 (SPY) NASDAQ (QQQ)	FTSE 100 (EZU)	Nikkei 225 (EWJ)
Technology	Apple (AAPL) Google (GOOGL) Amazon (AMZN) Facebook (FB) Microsoft (MSFT) Twitter (TWTR)	Nokia (NOK) Philips (PHIA.AS) Siemens (SIE.DE)	Sony (6758.T) Baidu (BIDU) Tencent (0700.HK) Alibaba (BABA)
Financial services	JPMorgan Chase (JPM)	HSBC (HSBC)	CCB (0939.HK)
Energy	ExxonMobil (XOM)	Shell (RDSA.AS)	PetroChina (PTR)
Automotive	Tesla (TSLA)	Volkswagen (VOW3.DE)	Toyota (7203.T)
Food	Coca Cola (KO)	AB InBev (ABI.BR)	Kirin (2503.T)

As far as the trading horizon is concerned, the 8 years preceding the original publication of this research work are assumed to be a relevant period of time. However, such a limited time period could (rightly) be criticised to not sufficiently be representative of the entire set of market phenomena. For instance, the financial crisis of 2008 is not taken into consideration, even though it could be interesting to assess the robustness of a trading strategy with respect to such an extraordinary event. This important choice is motivated by two reasons. Firstly, a shorter trading horizon is less likely to contain significant market regime shifts that could seriously harm the training stability of trading strategies. Secondly, the availability of financial data substantially drops when going further into the past. Finally, the 8-year trading horizon is divided into both training and test sets as follows:

- **Training set:** 01/01/2012 $\rightarrow$ 31/12/2017.
- **Test set:** 01/01/2018 $\rightarrow$ 31/12/2019.

Besides that, a validation set is selected as a subset of the training set, for the tuning of the numerous hyperparameters of the TDQN algorithm. It should also be specified that, after the training phase, the parameters of the algorithmic trading strategy are fixed during its evaluation on the test set, meaning that the new experiences acquired are not valued for extra training. However, this approach constitutes an interesting future research direction.

Eventually, there is still room for improvements with regard to the new benchmark stock markets introduced, by incorporating further diversification. Firstly, more stocks together with some instruments from other financial markets could undoubtedly cover complementary market phenomena that need to be addressed appropriately. Secondly, the time horizon of the trading activity could be further extended with the same objective, as previously explained.

3.5.2 Benchmark trading strategies

In order to accurately assess both the strengths and weaknesses of the TDQN algorithm, some benchmark algorithmic trading strategies are selected for comparison purposes. More precisely, the most popular conventional trading strategies that are commonly adopted in practice are taken into consideration. For the ease of analysis, more advanced approaches are discarded for now, such a comparison being left as future work. Although the TDQN algorithm mostly produces active trading strategies, both passive and active strategies are included within the benchmark. Naturally, all the trading strategies share the same input and output spaces presented in Section 3.3.4 ($\mathcal{O}$ and $\mathcal{A}$), for the sake of fairness. The following list summarises the benchmark trading strategies selected:

- **Buy and Hold** (BH).
- **Sell and Hold** (SH).
- **Trend Following with moving averages** (TF).
- **Mean Reversion with moving averages** (MR).

The first two benchmark trading strategies (BH and SH) are said to be *passive*, because there are no changes in trading position over the trading horizon. On the contrary, the other two benchmark strategies (TF and MR) are *active* trading strategies, that issue multiple changes in trading position over the trading horizon. On the one hand, a *trend following* trading strategy is concerned with the identification and the follow-up of significant market trends, as depicted in Figure 3.7. On the other hand, a *mean reversion* trading strategy, illustrated in Figure 3.8, exploits the tendency of the market to get back to its previous average price in the absence of clear trends. Therefore, by design, a trend following trading strategy generally makes a profit when a mean reversion trading strategy does not, with the opposite being true as well. Indeed, these two families of trading strategies adopt opposite positions: a mean reversion strategy always denies and goes against the trends while a trend following strategy follows the movements. For more detailed about these classical trading strategies, the reader can refer to the following documents [19, 35, 36].

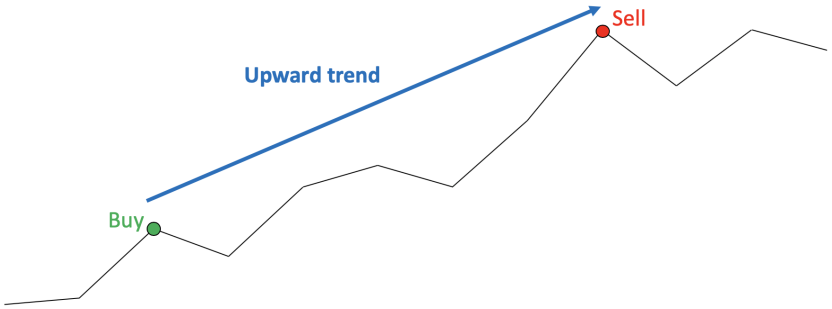


Figure 3.7: Illustration of a typical trend following trading strategy.

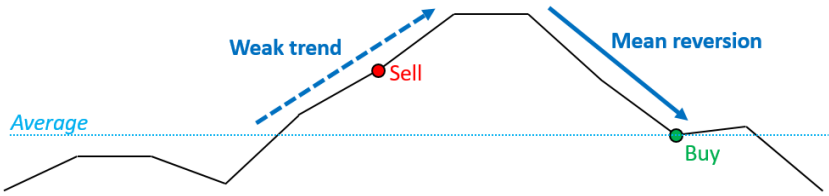


Figure 3.8: Illustration of a typical mean reversion trading strategy.

3.5.3 Quantitative performance assessment

In the scope of the algorithmic trading problem, the quantitative performance assessment consists in defining some performance indicators to quantify numerically the results achieved by a trading strategy. Because the core objective of an algorithmic trading strategy is to be profitable, its performance should be linked to the amount of money earned. However, as previously explained, this reasoning omits to take into consideration the risk associated with the trading activity, which should ideally be mitigated. There is a trade-off between raw performance and risk which depends on the investor's profile. Therefore, the quantitative performance indicators have to assess both aspects of the performance of a trading strategy.

Table 3.3: Quantitative performance assessment indicators adopted.

Performance indicator	**Description**
Sharpe ratio	Return of a trading activity compared to its riskiness.
Profit & Loss	Money gained or lost at the end of a trading activity.
Annualised return	Annualised return generated during a trading activity.
Annualised volatility	Modelling of the risk associated with a trading activity.
Profitability ratio	Percentage of winning trades made during a trading activity.
Profit and loss ratio	Ratio between the average profit and loss of the trades made.
Sortino ratio	Sharpe ratio with solely the negative risk being penalised.
Maximum drawdown	Largest loss from a peak to a trough during a trading activity.
Maximum drawdown duration	Time duration of the maximum drawdown of a trading activity.

To achieve that goal, multiple quantitative performance indicators have been selected. Previously introduced in Section 3.3.4, the most important one is certainly the Sharpe ratio. Indeed, this performance indicator, which is very popular in the algorithmic trading field, is particularly informative since it combines both profitability and risk into a single figure. In addition to the Sharpe ratio, this research work makes use of various other quantitative performance indicators in order to provide extra insights. Table 3.3 summarises the set of nine performance indicators adopted to quantify the performance of a trading strategy. For more details about these indicators, please refer to the following references [19, 35, 36].

Complementarily to the computation and analysis of these performance indicators, it is interesting to graphically represent the behaviour of a trading strategy. Plotting both the stock market price p_t and portfolio value v_t evolutions together with the trading actions a_t issued by the trading strategy seems appropriate to accurately analyse the trading policy learnt. Moreover, this visualisation could also provide extra insights about the performance, the strengths and the weaknesses of the strategy analysed.

3.6 Results and discussion

This section presents a thorough evaluation of the TDQN algorithm proposed, following the performance assessment methodology previously described in Section 3.5.3. Firstly, a detailed analysis is carried out both for a case that performs well and for a case where the results are mitigated at best. This perfectly highlights the strengths, weaknesses and limitations of the TDQN algorithm. Secondly, the performance achieved by this novel algorithmic solution on the complete benchmark is summarised and analysed. Finally, some additional discussions about the discount factor parameter, the influence of the trading costs as well as the main challenges faced by the DRL approach are provided. For the sake of reproducibility, the experimental code supporting the results presented in this thesis chapter is publicly available at the following link:

`https://github.com/ThibautTheate/An-Application-of-Deep-Reinforcement-Learning-to-Algorithmic-Trading`.

3.6.1 Positive results - Apple stock

The first detailed analysis carried out is related to the trading strategies generated by the TDQN algorithm for the Apple stock, for which promising results are reported. In practice, the DRL algorithms are generally characterised by a non-negligible variance, especially in the context of stochastic environments. The TDQN algorithm is no exception. Indeed, despite identical initial conditions, different training experiments inevitably lead to distinct trading policies whose performance slightly varies. For this reason, the analysis of the results achieved by the TDQN algorithm is performed in two steps. Firstly, a typical trading strategy learnt by the TDQN algorithm is inspected and compared to the benchmark trading strategies. Secondly, the expected performance of the TDQN algorithm is discussed.

Typical policy. Firstly, a representative trading policy learnt by the TDQN algorithm is quantitatively evaluated and compared to the benchmark trading strategies. Table 3.4 summarises this performance assessment, with an initial capital of \$100,000. The TDQN algorithm achieves really positive results from both raw performance and risk mitigation perspectives, clearly outperforming the benchmark active and passive trading strategies. Despite not being the best performing solution for all performance indicators, the TDQN algorithm is more balanced and always remains very close to the top results. Secondly, a qualitative and graphical analysis is carried out. Figure 3.9 plots both evolutions of the stock market price p_t and of the trading agent's portfolio value v_t, alongside the actions a_t outputted by the trading policy learnt via the TDQN algorithm. It can be observed that the RL agent is capable of accurately detecting and benefiting from dominant market trends, either upward or downward, while being less effective when the volatility increases during market behavioural shifts. Additionally, the trading agent generally lags slightly behind the market trends, meaning that the policy learnt by the TDQN algorithm for this particular stock market is more reactive than proactive. This behaviour can be expected with such a limited observation space $\mathcal{O}$ compared to the complexity of the environment, not including the reasons for the future market directions (news, financial reports, macroeconomics). Still, the trading strategy learnt is not purely reactive. Indeed, it can be observed that the RL agent may adapt its trading position slightly before the occurrence of a new trend inversion, on the basis of a sudden increase in volatility (anticipation).

Expected performance. Figure 3.10 plots the evolution during the training phase of the averaged performance achieved by the trading policies learnt via the TDQN algorithm for both the training and test sets, with the variance highlighted as a shaded area. The expected performance reported is in line with the results previously presented for a typical run of the TDQN algorithm, but the associated variance is non-negligible. Moreover, it can be observed from this figure that the problematic overfitting phenomenon is appropriately managed in this particular case. In fact, the performance for the test set being superior to that for the training set is not a mistake. This simply indicates a more profitable market in the test set compared to the training set. This observation perfectly illustrates a major difficulty associated with the algorithmic trading problem: the distinct distributions within the training and test sets. Indeed, the probability distribution of the daily returns is continuously changing, which significantly complicates both training and evaluation of the trading strategies.

Table 3.4: Quantitative performance assessment for the Apple stock.

Performance indicator	**BH**	**SH**	**TF**	**MR**	**TDQN**
Sharpe ratio	1.239	-1.593	1.178	-0.609	**1.484**
Profit & Loss [$]	79823	-80023	68738	-34630	**100288**
Annualised return [%]	28.86	-100.00	25.97	-19.09	**32.81**
Annualised volatility [%]	26.62	44.39	**24.86**	28.33	25.69
Profitability ratio [%]	**100**	0.00	42.31	56.67	52.17
Profit and loss ratio	∞	0.00	3.182	0.492	2.958
Sortino ratio	1.558	-2.203	1.802	-0.812	**1.841**
Max drawdown [%]	38.51	82.48	**14.89**	51.12	17.31
Max drawdown duration [days]	62	250	**20**	204	25

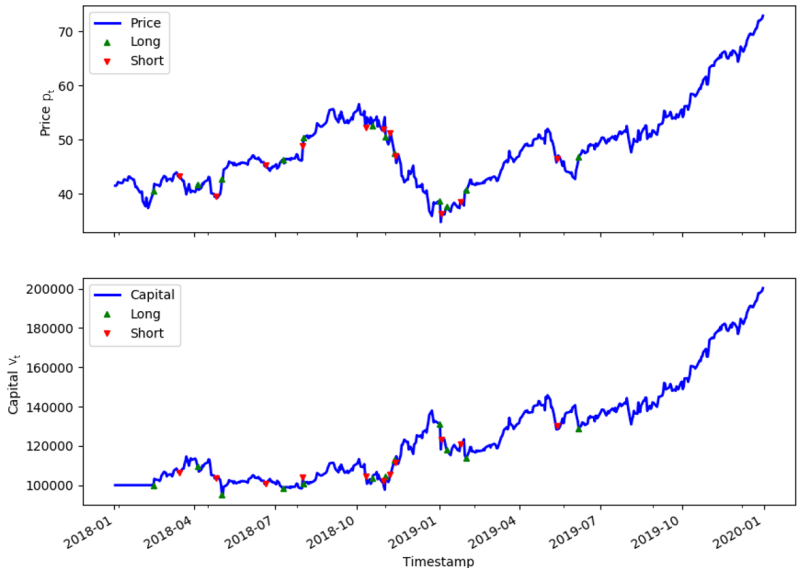


Figure 3.9: Behaviour of a typical trading strategy produced by the TDQN algorithm for the Apple stock.

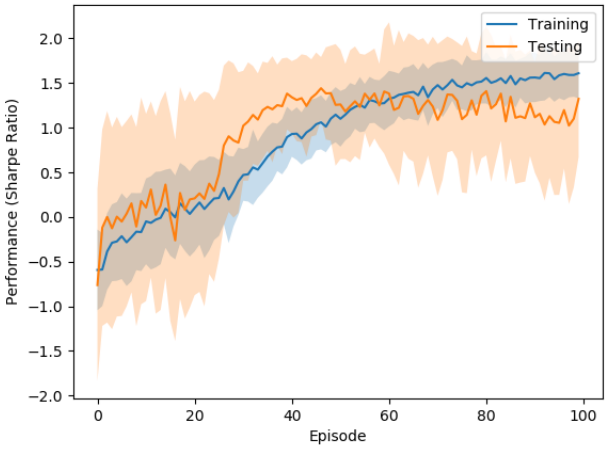


Figure 3.10: Expected performance of the trading strategies produced by the TDQN algorithm (Apple stock).

3.6.2 Mitigated results - Tesla stock

The same detailed analysis is carried out for the Tesla stock, which presents very different characteristics compared to the Apple stock, such as a pronounced volatility. In contrast to the promising performance reported for the previous stock, this case has been specifically selected to highlight the limitations of the TDQN algorithm. The intuition behind the choice of the Tesla stock is the following. Over the time period targeted, it could be demonstrated that the price of a Tesla share has been significantly impacted by the news, an information that is not taken into consideration by the RL agent (see observation space in Section 3.3.4). Consequently, the observability of the environment further decreases, which is expected to seriously harm the performance of the TDQN algorithm.

Typical policy. Table 3.5 presents the performance achieved by a typical trading policy learnt via the TDQN algorithm alongside the results from the benchmark trading strategies, with an initial capital of \$100,000. The mitigated performance achieved by the benchmark active trading strategies suggests that the Tesla stock is quite difficult to properly manage. As previously explained, it is likely caused by the significant volatility together with the lack of important information within the observation space. Even though the typical trading strategy produced by the TDQN algorithm achieves a positive Sharpe ratio, almost no profit is generated. Moreover, the risk level associated with this trading activity is undoubtedly not acceptable. For instance, both the maximum drawdown and its duration are particularly large, which would result in a stressful situation for the operator responsible for the trading strategy. Figure 3.11, plotting the evolution of both the stock market price p_t and the RL agent's portfolio value v_t together with the trading decisions a_t made, confirms the former observation. Additionally, this graph reveals that the typical trading strategy learnt via the TDQN algorithm presents a higher trading frequency. Despite the non-negligible trading costs, numerous changes in the trading position are performed, which further increases the riskiness of the trading activity.

Expected performance. Figure 3.12 plots the expected performance and variance of the trading policies learnt by the TDQN algorithm for both the training and test sets, as a function of the number of training episodes (training phase). The expected performance reported is substantially higher than the results achieved by the typical policy previously analysed. Consequently, the latter can not be considered as representative of the average behaviour, but has been selected for the ensuing discussion. This observation highlights a key limitation of the TDQN algorithm: the considerable variance that may potentially lead to the selection of a poorly performing policy compared to the expected performance. Then, another observation from this figure concerns the significantly higher performance achieved on the training set compared to the test set. This suggests that the DRL algorithm is subject to overfitting in this case, despite the multiple regularisation techniques implemented. Once again, this overfitting phenomenon can be partially explained by the observation space which is too limited to effectively apprehend the Tesla stock.

Table 3.5: Quantitative performance assessment for the Tesla stock.

Performance indicator	**BH**	**SH**	**TF**	**MR**	**TDQN**
Sharpe ratio	**0.508**	-0.154	-0.987	0.358	0.261
Profit & Loss [$]	**29847**	-29847	-73301	8600	98
Annualised return [%]	**24.11**	-7.38	-100.00	19.02	12.80
Annualised volatility [%]	53.14	**46.11**	52.70	58.05	52.09
Profitability ratio [%]	**100**	0.00	34.38	67.65	38.18
Profit and loss ratio	∞	0.00	0.534	0.496	1.621
Sortino ratio	**0.741**	-0.205	-1.229	0.539	0.359
Max drawdown [%]	**52.83**	54.09	79.91	65.31	58.95
Max drawdown duration [days]	205	**144**	229	159	331

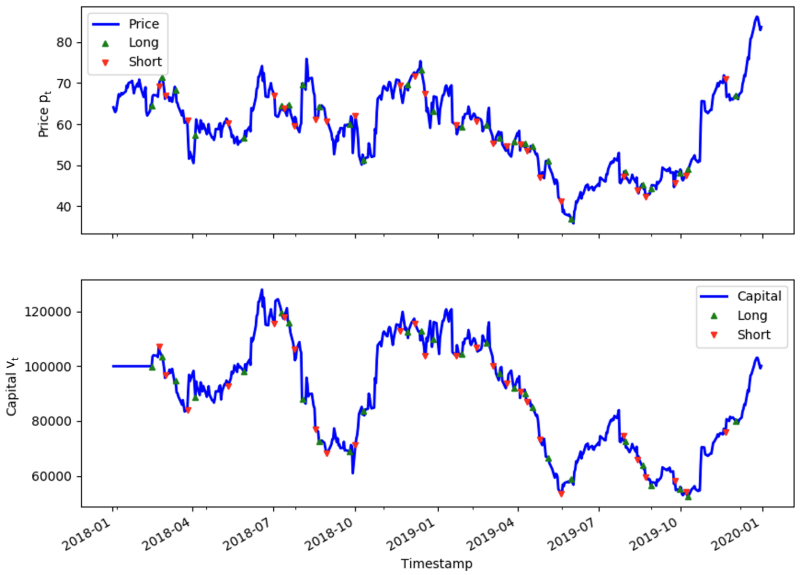


Figure 3.11: Behaviour of a typical trading strategy produced by the TDQN algorithm for the Tesla stock.

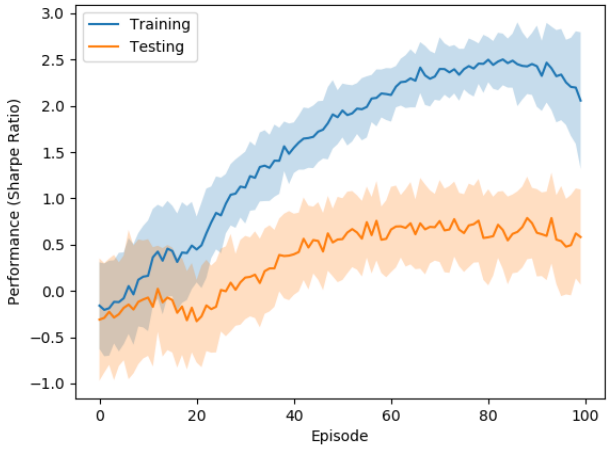


Figure 3.12: Expected performance of the trading strategies produced by the TDQN algorithm (Tesla stock).

3.6.3 Global results - Testbench

Despite being truly insightful regarding both the strengths and weaknesses of the TDQN algorithm, the two previous detailed analyses are not sufficient to draw robust conclusions. As previously explained, the new algorithmic trading solution proposed has to be rigorously evaluated on a set of diversified markets. To meet this requirement, the TDQN algorithm is quantitatively assessed on the benchmark introduced in Section 3.5.1, which is composed of 30 stocks. More precisely, Table 3.6 summarises the expected Sharpe ratio achieved by the trading policies learnt via the TDQN algorithm alongside the benchmark trading strategies detailed in Section 3.5.2, for all the 30 stocks taken into consideration.

To begin with, the performance achieved by the benchmark trading strategies is briefly discussed. It is important to differentiate the passive trading strategies (BH and SH) from the active ones (TF and MR). This second category has way more potential, at the cost of an extra non-negligible risk associated with the trading activity: continuous speculation. On the contrary, the passive trading strategies could be viewed as long-term investments, generally safer (provided a relevant choice is made for the initial trading position) but also less profitable. Having a closer look at the time period, it appears that the training and test sets present quite different distributions for the daily returns, which complicates the training of active strategies. The stock markets were mostly *bullish* (price p_t primarily increasing over time), with more volatility and instabilities during the trading period composing the test set. Therefore, it is not surprising to have the BH passive trading strategy outperforming the other benchmark strategies. In fact, both active trading strategies fail to generate positive results on average on the benchmark studied. This observation suggests that these market conditions are very complex to actively trade. Finally, this poorer performance can also be explained by the lack of versatility of such trading strategies which are designed to exploit specific market patterns, but are less effective in more diversified situation.

As far as the new TDQN algorithm is concerned, promising yet mixed results are reported. On the one hand, the proposed methodology significantly outperforms the benchmark active trading strategies. On the other hand, the TDQN algorithm barely surpasses the BH trading strategy, even though it has to be admitted that the particular market conditions of the test set are favourable to this simple passive strategy. Interestingly, the TDQN algorithm and BH strategy achieves an identical performance for certain stocks. This observation is explained by the ability of the TDQN algorithm to learn to switch to a passive trading strategy when uncertainty is too pronounced. Once again, the restrained observation space is key in making the market environment stochastic and uncertain. When the TDQN algorithm learns active trading strategies, it has been observed that both the *trend following* and *mean reversion* patterns can be effectively exploited, which is really encouraging. Finally, based on these results, the main advantage of the DRL approach for algorithmic trading is probably its versatility and ability to efficiently adapt to various markets presenting diverse characteristics.

Table 3.6: Quantitative performance assessment for the complete benchmark.

Stock	**Sharpe Ratio**				
	BH	*SH*	*TF*	*MR*	*TDQN*
Dow Jones (DIA)	**0.684**	-0.636	-0.325	-0.214	**0.684**
S&P 500 (SPY)	**0.834**	-0.833	-0.309	-0.376	**0.834**
NASDAQ 100 (QQQ)	**0.845**	-0.806	0.264	0.060	**0.845**
FTSE 100 (EZU)	0.088	0.026	-0.404	-0.030	**0.103**
Nikkei 225 (EWJ)	0.128	-0.025	-1.649	**0.418**	0.019
Google (GOOGL)	**0.570**	-0.370	0.125	0.555	0.227
Apple (AAPL)	1.239	-1.593	1.178	-0.609	**1.424**
Facebook (FB)	**0.371**	-0.078	0.248	-0.168	0.151
Amazon (AMZN)	**0.559**	-0.187	0.161	-1.193	0.419
Microsoft (MSFT)	**1.364**	-1.390	-0.041	-0.416	0.987
Twitter (TWTR)	0.189	**0.314**	-0.271	-0.422	0.238
Nokia (NOK)	-0.408	0.565	1.088	**1.314**	-0.094
Philips (PHIA.AS)	**1.062**	-0.672	-0.167	-0.599	0.675
Siemens (SIE.DE)	0.399	-0.265	0.525	**0.526**	0.426
Baidu (BIDU)	-0.699	**0.866**	-1.209	0.167	0.080
Alibaba (BABA)	**0.357**	-0.139	-0.068	0.293	0.021
Tencent (0700.HK)	-0.013	**0.309**	0.179	-0.466	-0.198
Sony (6758.T)	**0.794**	-0.655	-0.352	0.415	0.424
JPMorgan Chase (JPM)	0.713	-0.743	-1.325	-0.004	**0.722**
HSBC (HSBC)	-0.518	**0.725**	-1.061	0.447	0.011
CCB (0939.HK)	0.026	0.165	-1.163	-0.388	**0.202**
ExxonMobil (XOM)	0.055	**0.132**	-0.386	-0.673	0.098
Shell (RDSA.AS)	0.488	-0.238	-0.043	**0.742**	0.425
PetroChina (PTR)	-0.376	**0.514**	-0.821	-0.238	0.156
Tesla (TSLA)	**0.508**	-0.154	-0.987	0.358	0.621
Volkswagen (VOW3.DE)	0.384	-0.208	-0.361	**0.601**	0.216
Toyota (7203.T)	**0.352**	-0.242	-1.108	-0.378	0.304
Coca Cola (KO)	1.031	-0.871	-0.236	-0.394	**1.068**
AB InBev (ABI.BR)	-0.058	**0.275**	0.036	-1.313	0.187
Kirin (2503.T)	0.106	0.156	-1.441	0.313	**0.852**
Average	0.369	-0.202	-0.331	-0.056	**0.404**

3.6.4 Discussion about the discount factor

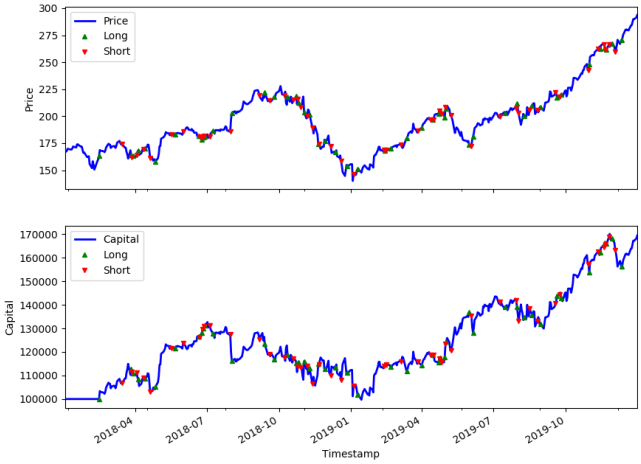
As previously explained in Section 3.3.4, the discount factor, denoted $\gamma \in [0, 1]$, is concerned with the relative importance of future rewards. Without going into detail, this parameter is generally set between 0.9 and 0.999 for most control problems solved with RL. However, in the scope of the algorithmic trading problem, the proper tuning of the discount factor is not trivial because of the considerable uncertainty assigned to the future. On the one hand, the desired trading policy should be long-term oriented ($\gamma \to 1$), in order to avoid a high trading frequency that would expose the agent to considerable trading costs. On the other hand, it would be unwise to place too much importance into an uncertain future ($\gamma \to 0$). Therefore, there intuitively exists a trade-off for the discount factor parameter.

This theoretical reasoning is validated by the multiple experiments performed to tune the parameter γ. Indeed, it has been observed that there is an optimal value for the discount factor, which is neither too small nor too large. Moreover, the experiments highlight another link between this important parameter and the trading frequency of the RL agent, because of the trading costs. From the perspective of the RL agent, these costs represent an obstacle to a change in trading position. Indeed, changing its trading position inevitably results in an immediate reduced reward for the agent. Therefore, the RL agent has to be sufficiently confident about the future in order to overcome the lower immediate reward caused by the trading costs. Since the discount factor sets the relative importance of the future, the value of this parameter has to be large enough to allow an active trading strategy to be learnt.

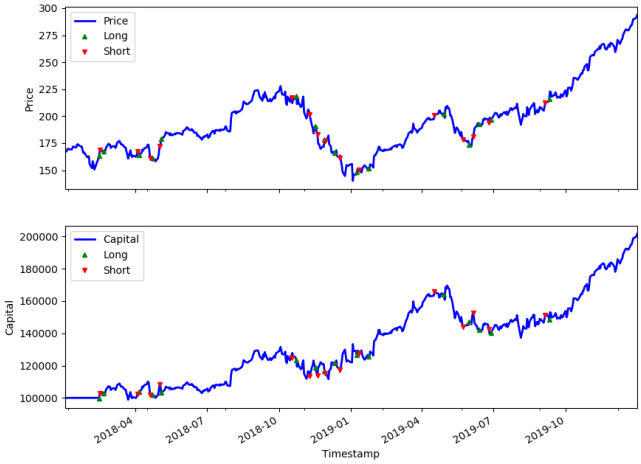
3.6.5 Discussion about the trading costs

As previously hinted, the trading costs play a major role in algorithmic trading. Indeed, the performance of a trading strategy may be strongly impacted by these costs, which should therefore be rigorously taken into consideration. A major motivation for investigating a DRL approach rather than pure forecasting methodologies that could also be based on innovative AI techniques is related to the trading costs. As previously explained in Section 3.3, the RL formalism allows the consideration of these costs directly into the decision-making process. Therefore, the trading policy is learnt according to the exact value assigned to the trading costs. On the contrary, a purely predictive approach based on DL techniques would learn to forecast the future market prices, but would not provide any indications about an appropriate trading strategy taking into account the trading costs. Although the former methodology offers more flexibility and can also lead to performing trading policies, it is less efficient by design because of these important trading costs.

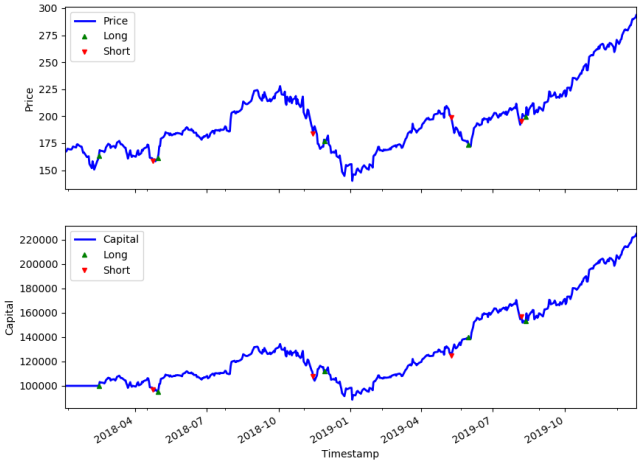
In order to illustrate the ability of the TDQN algorithm to adapt the produced policies according to the trading costs, Figure 3.13 presents the trading activities of three typical strategies learnt for three different values of the trading costs. It can be clearly observed from that figure that the trading frequency is reduced when the trading costs increase, as expected. Moreover, when the costs become too high, the TDQN algorithm learns to adopt a passive trading strategy (buy/sell and hold).



(a) Trading costs: 0%.



(b) Trading costs: 0.1%.



(c) Trading costs: 0.2%.

Figure 3.13: Impact of the trading costs on the trading strategies learnt by the TDQN algorithm, in the context of the Apple stock.

3.6.6 Challenges identified

Nowadays, the DRL approach has already become the state of the art for numerous real-world problems. In particular, this AI-based methodology excels in specific environments such as games, the most impressive success being the *AlphaGo* algorithm [70]. Nevertheless, in the scope of the algorithmic trading problem, the market environment presents very different characteristics: stochasticity, uncertainty and poor observability, among others. Naturally, numerous challenges remain in order to solve the sequential decision-making problem behind algorithmic trading with a DRL approach, the main ones being summarised hereafter.

Firstly, markets are stochastic environments requiring the algorithmic solution considered to be particularly robust to uncertainty. Moreover, the extremely poor observability of the market environments further reinforces this problematic uncertainty. From the perspective of the RL agent, the amount of information at its disposal may not be sufficient to properly understand the dynamics of the environment. If the agent is not able to accurately explain the market phenomena happening, the training will be ineffective. Secondly, the market environments are continuously changing and may even be subjected to significant regime shifts. In other words, the probability distributions of important quantities such as the daily returns are constantly evolving. However, ML approaches, including the RL methodology, require the training data (past) to be representative of the test data (future). For this reason, the continuous evolution of markets inevitably hurts the performance of DRL algorithms, by design. Thirdly, for most problems related to markets, a successful decision-making does not solely maximises the raw performance, but does also efficiently mitigate the associated risk. Although the risk can be somehow included within the design of the rewards, a more convenient solution would be to keep the rewards unchanged (about raw performance) and to take into consideration the risk directly within the RL algorithm. This research direction is being explored, but would definitively deserve more attention from the RL community. Fourthly, partly caused by the stochasticity and poor observability of the market environments, the RL approach may heavily suffer from the problematic overfitting phenomenon. As suggested in paper [14], there is a critical need for more rigorous evaluation protocols in RL because of the strong tendency of common DRL techniques to overfit. Additional research on this specific topic is required for the DRL approach to fit a broader range of real-world applications. Fifthly, the substantial variance of DRL techniques such as the popular DQN algorithm may hinder their application to certain decision-making problems, including those related to markets where there is generally a lot of money at stake. Lastly, the AI-based solutions, including the one presented in this research work, are commonly viewed as black-box models, that do not inspire trust to most users. For the RL approach to solve more real-world problems, the explainability of the decision-making process has to be improved.

On the basis of these conclusions, the doctoral thesis presents new scientific contributions to three of the major problems discussed. Chapter 4 introduces an important research about the distributional RL approach, which, to the author's opinion, may be particularly well suited to handle stochastic environments such as markets. Chapter 5 goes a step further by presenting a novel methodology to learn risk-sensitive policies based on the distributional RL approach, while improving the explainability of the decision-making process.

3.7 Conclusion

This thesis chapter presents a detailed formalisation of the algorithmic trading problem of determining the optimal trading position at any point in time during a trading activity in the stock markets. More precisely, this challenging sequential decision-making problem is cast as a RL problem. Additionally, this research work presents the *Trading Deep Q-Network* (TDQN) algorithm, an AI-based solution on the basis of the deep reinforcement learning (DRL) approach. Following a new quantitative performance assessment designed to promote more reliable results in algorithmic trading, promising results are reported for the TDQN algorithm, which outperforms the benchmark conventional trading strategies. Moreover, the proposed solution demonstrates key benefits compared to classical approaches, including an encouraging versatility as well a remarkable adaptability to various trading costs. Finally, the limitations of the TDQN algorithm are thoroughly discussed, highlighting the important research directions that are necessary to make the RL approach a viable solution to market environments in general.

This thesis chapter is concluded with several avenues that are suggested as future work, since there is room for improvements. Firstly and most importantly, the main limitations of the RL approach to the algorithmic trading problem and more generally for market environments, detailed in Section 3.6.6, should receive more attention for the research community. Secondly, some assumptions related to the formalisation of the sequential decision-making problem behind algorithmic trading could be reviewed. For instance, the observation space could be extended to enhance the observability of the market environment. Similarly, some constraints about the action space could be slightly relaxed in order to allow new possibilities for the agent. In addition, advanced RL reward engineering could be performed to narrow the gap between the RL objective and the Sharpe ratio maximisation objective. Thirdly, diverse improvements to the DRL algorithm could be investigated. For instance, including LSTM layers into the DNN would be a better fit for the financial time-series data [71]. Another important enhancement would be the implementation of the various improvements that are promoted by the Rainbow algorithm [56]: multi-step learning [57], double Q-learning [58], prioritised experience replay [59], duelling architecture [60], distributional RL [61] and noisy networks [62]. Finally, it would be interesting to investigate a DRL approach from the *policy optimisation* category, such as the *Proximal Policy Optimisation* (PPO) algorithm [72].

Progress is made by trial and failure; the failures are generally a hundred times more numerous than the successes, yet they are usually left unchronicled.

— William Ramsay

Chapter 4

Distributional Reinforcement Learning with Unconstrained Monotonic Neural Networks

Figure 4.1: Illustration of Chapter 4 entitled *Distributional Reinforcement Learning with Unconstrained Monotonic Neural Networks*, created by a generative art AI [1].

Chapter overview

The distributional reinforcement learning approach advocates for representing the complete probability distribution of the random return instead of solely modelling its expectation. A distributional RL algorithm may be characterised by two main components, namely the representation together with its parameterisation of the distribution and the probability metric defining the loss. The present research work considers the *unconstrained monotonic neural network* (UMNN) architecture, which is a universal approximator of continuous monotonic functions, particularly well suited for modelling different representations of a probability distribution. This property enables the efficient decoupling of the effect of the function approximator class from that of the probability metric. Firstly, the research work introduces a methodology for learning different representations of the random return probability distribution (PDF, CDF and QF). Secondly, a novel distributional RL algorithm named *unconstrained monotonic deep Q-network* (UMDQN) is presented. To the author's knowledge, it is the first distributional RL method supporting the learning of *three*, *valid* and *continuous* representations of the random return distribution. Lastly, in light of this new algorithm, an empirical comparison is performed between three probability quasi-metrics commonly employed in distributional RL, namely the Kullback-Leibler divergence, the Cramer distance and the Wasserstein distance. The results highlight the strengths and weaknesses associated with each probability metric, together with an important limitation of the Wasserstein distance.

This thesis chapter is primarily based on the following scientific publication [4]:

Thibaut Théate, Antoine Wehenkel, Adrien Bolland, Gilles Louppe and Damien Ernst. Distributional Reinforcement Learning with Unconstrained Monotonic Neural Networks. *Neurocomputing*, 534:199-219, 2023.

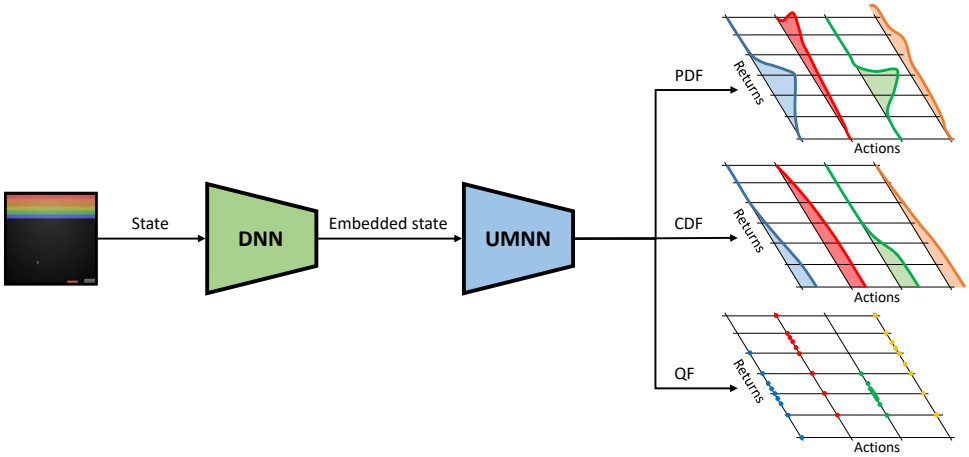


Figure 4.2: General illustration of the novel technical solution presented in this thesis chapter entitled *Distributional Reinforcement Learning with Unconstrained Monotonic Neural Networks*.

4.1 Introduction

As previously explained in this thesis, *Reinforcement learning* (RL) is a family of techniques belonging to the area of *machine learning* (ML), concerned with the learning process of an agent sequentially interacting within an environment and aiming to maximise the notion of cumulative reward. *Deep reinforcement learning* (DRL) extends this approach by employing *deep learning* (DL) techniques to generalise the information acquired from the interaction of the agent with its environment. Depending on whether a model of the environment is available and exploited or not, the RL algorithms can be classified *model-based* or *model-free*. The present research focuses exclusively on the second category, which can be subdivided into two classes: *policy optimisation* and *Q-learning*. The RL algorithms based on the *Q-learning* approach generally model the expectation of the random return, to ideally be maximised [51]. Alternatively, the *distributional RL* approach suggests learning the probability distribution of the random return. This methodology presents key advantages including learning richer representations of the returns collected, which leads to more efficient and stable learning, as well as making risk-sensitive control and exploration policies possible [73, 74].

A distributional RL algorithm may generally be characterised by two main components. The first one relates to both the representation together with its parameterisation of the random return probability distribution. A unidimensional distribution possesses several representations, such as the probability density function (PDF), cumulative distribution function (CDF) and quantile function (QF). Typically, *deep neural networks* (DNNs) are considered for approximating these various functions. The second important component concerns the probability quasi-metric adopted for comparing two distributions. Multiple quasi-metrics do exist for that purpose, the main ones experimented in distributional RL being the Kullback-Leibler (KL) divergence, the Cramer distance (which is also named energy distance), and the Wasserstein distance. In the rest of this thesis chapter, they will simply be referred to as *probability metrics*. In the context of distributional RL, the role of the probability metric is to quantitatively compare two distributions of the random return in order to implement a *temporal difference* (TD) learning method, similarly to the mean squared error between Q-values in classical RL. The choice of the probability metric is particularly important since each metric offers different theoretical convergence guarantees for distributional RL.

The core idea of this research work is to consider the *unconstrained monotonic neural network* (UMNN) architecture [75] in the scope of distributional RL. Originally designed for autoregressive flows, this particular architecture is in fact a universal approximator of continuous monotonic functions. Several works have already demonstrated the ability of this neural network to accurately model continuous monotonic functions in practice [76, 77]. Since both the CDF and QF are monotonic, the UMNN architecture is expected to offer superior capability compared to classical neural networks when it comes to representing distributions. Moreover, the PDF can also be efficiently represented by this architecture, when standing at the heart of a *normalizing flow* [78], by taking advantage of the *change of variables theorem* [75]. Because the UMNN architecture can effectively model different distribution representations, it enables the efficient decoupling of the effect of the function approximator class from that of the probability metric, making a fair comparison between probability metrics possible.

This leads to the main contributions of the present research work, which are threefold. Firstly, a methodology for learning three representations of the random return probability distribution, namely the PDF, CDF and QF, is introduced. Secondly, a novel distributional RL algorithm, named *unconstrained monotonic deep Q-network* (UMDQN), is presented. Basically, it combines the UMNN architecture with the previous methodology for learning three different valid representations of the continuous distribution of the random return. Thirdly, taking advantage of this new algorithm, the research work proposes an empirical comparison of three probability metrics commonly used in distributional RL, namely the KL divergence, the Cramer distance and the Wasserstein distance. This analysis highlights the main strengths and weaknesses associated with each probability metric, but also reveals an important limitation of the Wasserstein distance. Actually, the latter observation highlights a critical approximation made by several state-of-the-art distributional RL algorithms, that leads to the learning of inaccurate probability distributions for the random return. To the author's knowledge, the proposed algorithmic solution is the first distributional RL approach supporting the learning of *several* (PDF, CDF and QF), *valid* (by ensuring monotonicity) and *continuous* (as opposed to discrete) representations of the random return distribution. To end this introductory section, it should be emphasised that the core objective of this research work is not to present a novel distributional RL algorithm outperforming the state-of-the-art algorithms on a given testbench, typically the Atari-57 benchmark [53], but rather to empirically derive new insights about distributional RL from this algorithmic solution.

4.2 Literature review

Q-learning is a popular *model-free* RL approach which is concerned with the learning of the quantity Q representing the quality of executing a certain action in a particular state [51]. Originally based on tabular or linear approximations, the methodology has been significantly extended with the DQN algorithm [55] using a DNN for approximating the quantity Q in a non-linear setting. The next important evolution is undoubtedly the distributional RL approach advocating for learning the entire probability distribution of the random return instead of only modelling its expectation [73]. Fundamental research on distributional RL is still in its early stages, but key benefits have already been discovered [74, 79].

Multiple successful distributional RL algorithms can be found in the scientific literature, based on diverse representations of the random return distribution and different probability metrics. First of all, the *categorical DQN* (CDQN) algorithm [73], which is also known as *C51*, approximates the PDF of the random return through categorical distributions and uses the KL divergence for quantitatively comparing these distributions. The link between this pioneer distributional RL algorithm and the Cramer distance probability metric was later highlighted [80]. Alternatively, the *quantile regression DQN* (QR-DQN) algorithm [81] learns the distribution of the random return by manipulating the QF with fixed uniform quantile fractions and the Wasserstein distance. Compared to the CDQN algorithm, this approach has the advantage of avoiding the specification of a fixed support for the random return values. Nevertheless, both algorithms suffer from the same drawback of estimating the distribution

of the random return on fixed locations (either value or probability), with as a consequence that the distributions learnt are discrete. The *implicit quantile network* (IQN) algorithm [82] solves this problem by learning the quantile values from quantile fractions sampled from a uniform distribution $\mathcal{U}([0,1])$. This is achieved with a specific DNN representing the QF by mapping quantile fractions to quantile values and trained by minimising the Wasserstein distance. Finally, the *fully parameterised quantile function* (FQF) algorithm [83] extends the previous methodology by parameterising both quantile fraction and value axes. To do so, two DNNs are used: one for generating appropriate quantile fractions and one for mapping these quantile fractions to quantile values. They are jointly trained by minimising the Wasserstein distance once again. Table 4.1 summarises the key characteristics of these state-of-the-art distributional RL algorithms, and Figure 4.3 illustrates these important algorithms in the context of the Atari-57 benchmark [53].

Table 4.1: Key characteristics (representation of the random return probability distribution and probability metric) of the main state-of-the-art distributional RL algorithms.

Algorithm	**Probability distribution representation**	**Probability metric**
DQN	Expectation (non distributional RL)	L1 metric
CDQN	Categorical PDF (fixed discrete support)	KL divergence
QR-DQN	Discrete QF (fixed quantile fractions)	Wasserstein distance
IQN	Continuous QF (quantiles drawn from $\mathcal{U}([0,1])$)	Wasserstein distance
FQF	Continuous QF (quantiles sampled by a DNN)	Wasserstein distance

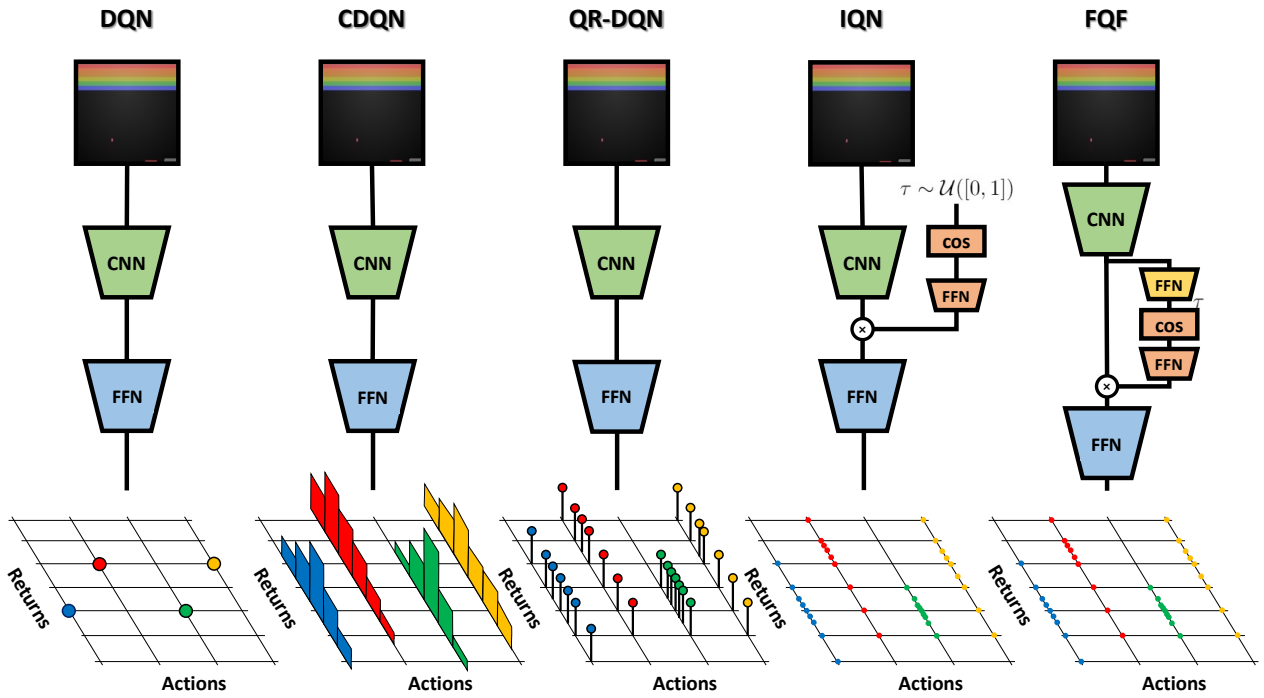


Figure 4.3: Illustration of the state-of-the-art distributional RL algorithms for the Atari-57 benchmark [53].

Besides the previous distributional RL algorithms that are well-established in the research community, one can mention several recent research works bringing new interesting insights about distributional RL. To begin with, the *moment matching DQN* (MMDQN) algorithm [84] learns via a DNN a finite set of statistics for the distribution of the random return by implicitly matching all orders of moments between the random return distribution and its target. The key benefit of this approach is to avoid the predefined statistic principle used in prior distributional RL works, leading to a simpler objective amenable to backpropagation. This is achieved by learning unrestricted statistics, i.e. deterministic samples, of the random return distribution by leveraging the maximum mean discrepancy technique from hypothesis testing. Then, sharing a similar philosophy to the present research work, the *non-crossing QR-DQN* algorithm [85] is an improvement of the well-established QR-DQN algorithm by implementing non-crossing quantile regression to ensure the monotonicity constraint for the QF. This enhancement is built on the observation that the non-decreasing property of learnt quantile curves is not guaranteed, which can lead to abnormal distribution estimates and reduce model interpretability. Nevertheless, this technique is not directly transferable to the IQN or FQF algorithms. To end this literature review, an important study reveals that the reward system of the human brain would operate similarly to distributional RL [86]. Indeed, the findings suggest that the human brain represents possible future rewards as a complete probability distribution and not as a single mean of stochastic outcomes. This is naturally very encouraging news supporting the soundness of the distributional RL approach.

4.3 Distributional Reinforcement Learning

This thesis adopts the standard RL setting where the agent interacts with its environment modelled as a *Markov decision process* (MDP). An MDP is a 6-tuple $(\mathcal{S},\ \mathcal{A},\ p_R,\ p_T,\ p_0,\ \gamma)$ where $\mathcal{S}$ and $\mathcal{A}$ respectively are the state and action spaces, $p_R(r|s,a)$ is the probability distribution from which the reward $r \in \mathbb{R}$ is drawn given a state-action pair (s,a), $p_T(s'|s,a)$ is the transition probability distribution, $p_0(s_0)$ is the probability distribution over the initial states $s_0 \in \mathcal{S}$, and $\gamma \in [0,1[$ is the discount factor. The RL agent makes decisions according to its policy $\pi : \mathcal{S} \to \mathcal{A}$, which is considered deterministic in the rest of this thesis chapter, mapping the states $s \in \mathcal{S}$ to the actions $a \in \mathcal{A}$.

As previously explained, the Q-learning approach focuses on modelling the *state-action value function* Q^π of a policy π. This quantity represents the expected discounted sum of rewards to be obtained by executing an action a in a state s and then following a policy π. Also named *expected return*, it satisfies the *Bellman equation* [52]:

$$Q^\pi(s,a) = \mathop{\mathbb{E}}_{s_t,r_t}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right], \qquad (s_0,a_0) := (s,a),\ a_t = \pi(s_t)\ , \tag{4.1}$$

$$Q^\pi(s,a) = \mathop{\mathbb{E}}_{s',r}\left[r + \gamma Q^\pi(s',\pi(s'))\right]. \tag{4.2}$$

Following this methodology, one can define the *optimal policy* π^* on the basis of the *optimal state-action value function* Q^* as the following:

$$Q^*(s,a) = \mathop{\mathbb{E}}_{s',r}\left[r + \gamma \max_{a' \in \mathcal{A}} Q^*(s',a')\right], \tag{4.3}$$

$$\pi^*(s) \in \mathop{\text{argmax}}_{a \in \mathcal{A}} Q^*(s,a) \ . \tag{4.4}$$

Distributional RL aims at modelling the entire probability distribution over returns instead of only its expectation. To this end, let the reward $R(s,a)$ be a random variable distributed under $p_R(\cdot|s,a)$, the *state-action value distribution* $Z^\pi \in \mathcal{Z}$ (which is also called *state-action return distribution function*) of a policy π is a random variable defined as follows:

$$Z^\pi(s,a) \overset{D}{=} \sum_{t=0}^{\infty} \gamma^t R(s_t,a_t) \ , \qquad (s_0,a_0) := (s,a),\ a_t = \pi(s_t),\ s_{t+1} \sim p_T(\cdot|s_t,a_t) \ , \tag{4.5}$$

where $A \overset{D}{=} B$ denotes the equality in distribution between the random variables A and B. Therefore, the state-action value function Q^π is the expectation of the *random return* Z^π. In a similar way, there exists a *distributional Bellman equation* recursively describing Z^π:

$$Z^\pi(s,a) \overset{D}{=} R(s,a) + \gamma P^\pi Z^\pi(s,a) \ , \tag{4.6}$$

$$P^\pi Z^\pi(s,a) :\overset{D}{=} Z^\pi(s',a') \ , \qquad s' \sim p_T(\cdot|s,a),\ a' = \pi(s') \ , \tag{4.7}$$

where $P^\pi : \mathcal{Z} \to \mathcal{Z}$ is the transition operator based on the decision-making policy π. Finally, one can define the *distributional Bellman operator* $\mathcal{T}^\pi : \mathcal{Z} \to \mathcal{Z}$ and the *distributional Bellman optimality operator* $\mathcal{T}^* : \mathcal{Z} \to \mathcal{Z}$ as the following:

$$\mathcal{T}^\pi Z^\pi(s,a) \overset{D}{=} R(s,a) + \gamma P^\pi Z^\pi(s,a) \ , \tag{4.8}$$

$$\mathcal{T}^* Z^*(s,a) \overset{D}{=} R(s,a) + \gamma Z^*\left(s', \pi^*(s')\right), \qquad s' \sim p_T(\cdot|s,a) \ . \tag{4.9}$$

Theoretically, the distributional Bellman operator $\mathcal{T}^\pi$ may potentially be a contraction mapping or not depending on the probability metric. This property implies that there exists a unique fixed point Z^π to converge towards when repeatedly applying the operator $\mathcal{T}^\pi$. For the distributional Bellman optimality operator $\mathcal{T}^*$, another condition is required for this contraction mapping property to hold: the optimal policy π^* has to be unique [73]. Multiple probability metrics do exist for quantitatively comparing the probability distributions of two continuous random variables. In this research work, the emphasis is set on the three main probability metrics used in distributional RL, namely the KL divergence, Cramer distance and Wasserstein distance. Table 4.2 formally introduces these probability metrics, together with their impact on the contraction mapping property of the distributional Bellman operator $\mathcal{T}^\pi$.

Table 4.2: Formal definition of the probability metrics studied, where A and B are two random variables, and where p_D, F_D and F_D^{-1} denote the PDF, CDF and QF of the random variable D, respectively.

Probability metric	**Mathematical definition**	$\mathcal{T}^\pi$ **contraction?**
KL divergence	$\mathcal{L}_{KL}(A,B) = D_{KL}(p_A, p_B) = \int_{\mathbb{R}} p_A(x) \log\left(\frac{p_A(x)}{p_B(x)}\right) dx$	No [87]
Cramer distance	$\mathcal{L}_C(A,B) = D_C(F_A, F_B) = \left(\int_{\mathbb{R}} \left(F_A(x) - F_B(x)\right)^2 dx\right)^{1/2}$	Yes [80]
Wasserstein distance	$\mathcal{L}_W(A,B) = D_W(F_A^{-1}, F_B^{-1}) = \int_0^1 F_A^{-1}(x) - F_B^{-1}(x) dx$	Yes [73]

4.4 Unconstrained monotonic deep Q-network

4.4.1 Learning different representations of a probability distribution

This section presents a new methodology for learning three different representations of the probability distribution of the random return: the PDF, CDF and QF. The learning process is based on the comparison of the left- and right-hand sides of the distributional Bellman equation (4.6). For a given probability metric $\mathcal{L}$, the random return Z^π is a fixed point of the Bellman operator $\mathcal{T}^\pi$ if it minimises the following loss:

$$\mathcal{L}\left(\mathcal{T}^\pi Z^\pi(s,a),\ Z^\pi(s,a)\right), \tag{4.10}$$

for all state-action pairs $(s,a) \in \mathcal{S} \times \mathcal{A}$. The distributional RL problem at hand will be addressed by defining a hypothesis space for the quantity Z^π and by minimising the loss function defined in Equation (4.10) over this space using *stochastic gradient descent* (SGD). In the following, the effect of the distributional Bellman operator $\mathcal{T}^\pi$ on the different representations of the random return probability distribution is rigorously studied. Intuitively, the discount factor γ *squeezes* the random return distribution while the reward R *shifts* this probability distribution, as illustrated in Figure 4.4 in the simplified particular situation of both deterministic reward and transition functions.

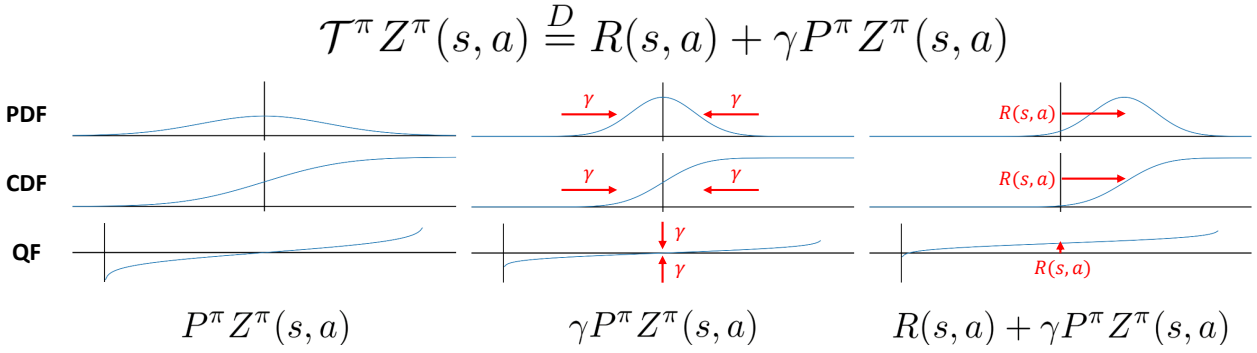


Figure 4.4: Illustration of the effect of the distributional Bellman operator on three different representations of the random return probability distribution in the simplified situation of both a deterministic reward function and a deterministic transition function.

PDF representation. Let $p_{Z^\pi}(z|s,a)$ be the PDF of the random variable Z^π given the state-action pair (s,a) at the return z. Assuming the KL divergence $\mathcal{L}_{KL}$ as the probability metric considered, the loss to be minimised defined in Equation (4.10) can be re-expressed as follows:

$$\mathcal{L}_{KL}(\mathcal{T}^\pi Z^\pi(s,a),\ Z^\pi(s,a)) = D_{KL}\left(p_{\mathcal{T}^\pi Z^\pi}(z|s,a),\ p_{Z^\pi}(z|s,a)\right) \tag{4.11}$$

$$= D_{KL}\left(\mathop{\mathbb{E}}_{s',r}\left[\frac{1}{\gamma}\, p_{Z^\pi}\left(\frac{z-r}{\gamma}\middle| s', \pi(s')\right)\right], p_{Z^\pi}(z|s,a)\right). \tag{4.12}$$

CDF representation. Let $F_{Z^\pi}(z|s,a)$ be the CDF of the random variable Z^π conditioned by the state-action pair (s,a) at the return z. Assuming the Cramer distance $\mathcal{L}_C$ as the probability metric considered, the loss formally defined in Equation (4.10) can be re-expressed as the following:

$$\mathcal{L}_C(\mathcal{T}^\pi Z^\pi(s,a),\ Z^\pi(s,a)) = D_C\left(F_{\mathcal{T}^\pi Z^\pi}(z|s,a),\ F_{Z^\pi}(z|s,a)\right) \tag{4.13}$$

$$= D_C\left(\mathop{\mathbb{E}}_{s',r}\left[F_{Z^\pi}\left(\frac{z-r}{\gamma}\middle| s', \pi(s')\right)\right], F_{Z^\pi}(z|s,a)\right). \tag{4.14}$$

QF representation. Let $F^{-1}_{Z^\pi}(\tau|s,a)$ be the QF of the random variable Z^π given the state-action pair (s,a) at the quantile fraction $\tau \in [0,1]$. Assuming the Wasserstein distance $\mathcal{L}_W$ as the probability metric considered, the loss to be minimised defined in Equation (4.10) can be re-expressed as follows:

$$\mathcal{L}_W(\mathcal{T}^\pi Z^\pi(s,a),\ Z^\pi(s,a)) = D_W\left(F^{-1}_{\mathcal{T}^\pi Z^\pi}(\tau|s,a),\ F^{-1}_{Z^\pi}(\tau|s,a)\right) \tag{4.15}$$

$$\simeq D_W\left(\mathop{\mathbb{E}}_{s',r}\left[r + \gamma F^{-1}_{Z^\pi}(\tau|s', \pi(s'))\right], F^{-1}_{Z^\pi}(\tau|s,a)\right). \tag{4.16}$$

As far as mathematical proofs are concerned, Equations (4.12) and (4.14) are respectively supported by Proposition 1 and Corollary 1 hereafter. On the contrary, Equation (4.16) could not be rigorously proven as originally intended. In order to get a better understanding of the challenge faced, some basic experiments have been conducted. The results suggest that Equation (4.16) results from an approximation of $F^{-1}_{\mathcal{T}^\pi Z^\pi}(\tau|s,a)$, leading to a random variable with the correct expectation but potentially different higher-order moments. In the scope of distributional RL, such an approximation may have two completely different implications depending on the objective pursued. If the intention is to accurately learn the probability distribution of the random return for implementing risk-aware policies, this approximation is obviously problematic. On the contrary, if the goal is to learn policies maximising the expectation of the random return, this approximation may have no negative effect since the distribution learnt has the correct first-order moment. In fact, this approach is adopted by the state-of-the-art QR-DQN, IQN and FQF algorithms which are able to learn valuable policies in practice, based on the expectation of the random return alone [81, 82, 83].

Proposition 1 Let $Z^\pi \in \mathcal{Z}$ be the random return associated with the policy $\pi : \mathcal{S} \to \mathcal{A}$, which is a random variable mapping the state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$ to the realisation of the return $z \in \mathbb{R}$. Additionally, let $p_R(r|s, a)$ be the probability distribution from which the reward $r \in \mathbb{R}$ is drawn, and $p_T(s'|s, a)$ be the transition probability distribution. Finally, let $\mathcal{T}^\pi : \mathcal{Z} \to \mathcal{Z}$ be the distributional Bellman operator, and let $Z^{\pi'} \in \mathcal{Z}$ be a random variable such that $Z^\pi = \mathcal{T}^\pi Z^{\pi'}$. Then, the probability density functions p_{Z^π} and $p_{Z^{\pi'}}$ associated with the random variables Z^π and $Z^{\pi'}$ respect the following equality:

$$p_{Z^\pi}(z|s, a) = \mathop{\mathbb{E}}_{\substack{r\sim p_R(\cdot|s,a) \\ s'\sim p_T(\cdot|s,a)}} \left[\frac{1}{\gamma}\, p_{Z^{\pi'}}\left(\frac{z-r}{\gamma} \middle| s', a' \right) \Big|_{a'=\pi(s')} \right] \quad \forall z \in \mathbb{R},\ s \in \mathcal{S},\ a \in \mathcal{A}\ . \tag{4.17}$$

Proof Let z be the return sampled from the random variable $Z^\pi(s, a)$ for the state-action pair (s, a). By marginalising over the reward r collected and over the next state-action pair (s', a') with $a' = \pi(s')$, the PDF of the random return can be expressed as follows:

$$p_{Z^\pi}(z|s, a) = \int p_{Z^\pi}(z|s, a, r, s', a')\ p(r, s', a'|s, a)\ dr\ ds'. \tag{4.18}$$

Considering both the conditional independence and the Markov property of the decision-making process, the expression $p(r, s', a'|s, a)$ can be re-written as the following:

$$p(r, s', a'|s, a) = p_R(r|s, a)p_T(s'|s, a). \tag{4.19}$$

According to the distributional Bellman equation, the return z can be expressed as a function of both the reward r and the next return z':

$$z = r + \gamma z'. \tag{4.20}$$

Based on this expression and making use of the change of variables theorem, the PDF $p_{Z^\pi}(z|s, a, r, s', a')$ can be re-expressed as follows:

$$p_{Z^\pi}(z|s, a, r, s', a') = |\gamma|^{-1}\, p_{Z^{\pi'}}\left(z'|s, a, r, s', a'\right) \big|_{z'=\frac{z-r}{\gamma}} \tag{4.21}$$

$$= \frac{1}{\gamma}\, p_{Z^{\pi'}}\left(\frac{z-r}{\gamma} \middle| s', a' \right). \tag{4.22}$$

Finally, by substitution of Equations (4.19) and (4.22) into Equation (4.18), the following relation is obtained:

$$p_{Z^\pi}(z|s, a) = \int \frac{1}{\gamma}\, p_{Z^{\pi'}}\left(\frac{z-r}{\gamma} \middle| s', a' \right) p_R(r|s, a)p_T(s'|s, a)\ dr\ ds' \tag{4.23}$$

$$= \mathop{\mathbb{E}}_{\substack{r\sim p_R(\cdot|s,a) \\ s'\sim p_T(\cdot|s,a)}} \left[\frac{1}{\gamma}\, p_{Z^{\pi'}}\left(\frac{z-r}{\gamma} \middle| s', a' \right) \Big|_{a'=\pi(s')} \right]. \tag{4.24}$$

□

Corollary 1 Let $Z^\pi \in \mathcal{Z}$ be the random return associated with the policy $\pi : \mathcal{S} \to \mathcal{A}$, which is a random variable mapping the state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$ to the realisation of the return $z \in \mathbb{R}$. Additionally, let $p_R(r|s, a)$ be the probability distribution from which the reward $r \in \mathbb{R}$ is drawn, and $p_T(s'|s, a)$ be the transition probability distribution. Finally, let $\mathcal{T}^\pi : \mathcal{Z} \to \mathcal{Z}$ be the distributional Bellman operator, and let $Z^{\pi'} \in \mathcal{Z}$ be a random variable such that $Z^\pi = \mathcal{T}^\pi Z^{\pi'}$. Then, the cumulative distribution functions F_{Z^π} and $F_{Z^{\pi'}}$ associated with the random variables Z^π and $Z^{\pi'}$ respect the following equality:

$$F_{Z^\pi}(z|s, a) = \mathop{\mathbb{E}}_{\substack{r\sim p_R(\cdot|s,a) \\ s'\sim p_T(\cdot|s,a)}} \left[F_{Z^{\pi'}} \left(\frac{z - r}{\gamma} \middle| s', a' \right) \Big|_{a'=\pi(s')} \right] \quad \forall z \in \mathbb{R},\ s \in \mathcal{S},\ a \in \mathcal{A}. \tag{4.25}$$

Proof By considering the definition of the CDF together with Equation (4.17) given by Proposition 1, the following development can be obtained:

$$\begin{aligned}
F_{Z^\pi}(z|s, a) &= \int_{-\infty}^{z} p_{Z^\pi}(z^*|s, a)\, dz^* && (4.26)\\
&= \int_{-\infty}^{z} \mathop{\mathbb{E}}_{\substack{r\sim p_R(\cdot|s,a) \\ s'\sim p_T(\cdot|s,a)}} \left[\frac{1}{\gamma}\, p_{Z^{\pi'}} \left(\frac{z^* - r}{\gamma} \middle| s', a' \right) \Big|_{a'=\pi(s')} \right] dz^* && (4.27)\\
&= \mathop{\mathbb{E}}_{\substack{r\sim p_R(\cdot|s,a) \\ s'\sim p_T(\cdot|s,a)}} \left[\int_{-\infty}^{z} \frac{1}{\gamma}\, p_{Z^{\pi'}} \left(\frac{z^* - r}{\gamma} \middle| s', a' \right) \Big|_{a'=\pi(s')} dz^* \right] && (4.28)\\
&= \mathop{\mathbb{E}}_{\substack{r\sim p_R(\cdot|s,a) \\ s'\sim p_T(\cdot|s,a)}} \left[\int_{-\infty}^{\frac{z-r}{\gamma}} p_{Z^{\pi'}} \left(z^{**} \middle| s', a' \right) \Big|_{a'=\pi(s')} dz^{**} \right] && (4.29)\\
&= \mathop{\mathbb{E}}_{\substack{r\sim p_R(\cdot|s,a) \\ s'\sim p_T(\cdot|s,a)}} \left[F_{Z^{\pi'}} \left(\frac{z - r}{\gamma} \middle| s', a' \right) \Big|_{a'=\pi(s')} \right]. && (4.30)
\end{aligned}$$

□

As previously mentioned, the case of the QF is more complex and involves an important approximation. Let $Z^\pi \in \mathcal{Z}$ be the random return associated with the policy $\pi : \mathcal{S} \to \mathcal{A}$, which is a random variable mapping the state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$ to the realisation of the return $z \in \mathbb{R}$. Additionally, let $p_R(r|s, a)$ be the probability distribution from which the reward $r \in \mathbb{R}$ is drawn, and $p_T(s'|s, a)$ be the transition probability distribution. Finally, let $\mathcal{T}^\pi : \mathcal{Z} \to \mathcal{Z}$ be the distributional Bellman operator, and let $Z^{\pi'} \in \mathcal{Z}$ be a random variable such that $Z^\pi = \mathcal{T}^\pi Z^{\pi'}$. Then, the quantile functions $F^{-1}_{Z^\pi}$ and $F^{-1}_{Z^{\pi'}}$ associated with the random variables Z^π and $Z^{\pi'}$ can be linked based on an approximation as the following:

$$F^{-1}_{Z^\pi}(\tau|s, a) \simeq \mathop{\mathbb{E}}_{\substack{r\sim p_R(\cdot|s,a) \\ s'\sim p_T(\cdot|s,a)}} \left[r + \gamma F^{-1}_{Z^{\pi'}}(\tau|s', a') \big|_{a'=\pi(s')} \right] \quad \forall \tau \in [0, 1],\ s \in \mathcal{S},\ a \in \mathcal{A}. \tag{4.31}$$

A limited empirical research on this approximation suggests that it leads to a random variable modelling the quantity Z^π with the correct expectation but potentially different higher-order moments. Moreover, the error resulting from this approximation is observed to increase with the stochasticity characterising the dynamics of the MDP (transition and reward probability distributions p_T and p_R). On the contrary, Equation (4.31) no longer relies on an approximation in the deterministic case. Consequently, this particular approximation may have two completely different implications depending on the objective pursued. If the distributional RL algorithm is used to learn ordinary decision-making policies maximising the expectation of the random return, the approach remains totally sound since the probability distribution learnt has the correct first-order moment. On the contrary, this approximation becomes really problematic if the intention is to learn the complete probability distribution of the random return for implementing risk-sensitive policies, as studied in Chapter 5.

4.4.2 Unconstrained monotonic neural network

As previously hinted, the PDF, CDF and QF of continuous random variables share the key property of being effectively modelled with strictly monotonic functions. This is the primarily reason for this research to consider *unconstrained monotonic neural networks* (UMNNs), which are universal approximators of continuous monotonic functions, for parameterising the random return probability distribution. Formally, a UMNN defines a parametric continuous monotonic function $G(\cdot;\theta) : \mathbb{R} \to \mathbb{R}$ as the following:

$$G(x;\theta) := \int_0^x g(t;\theta)dt + \beta \ , \tag{4.32}$$

where $g(\cdot;\theta) : \mathbb{R} \to \mathbb{R}^+$ is a free-form neural network whose output positiveness is enforced via an appropriate activation function (for instance ReLU or exponential), where θ denotes its parameters, and where $\beta \in \mathbb{R}$ is a trainable scalar parameter. This parameterisation can efficiently generalise to random variables conditioned by other quantities, in this case the state s and the action a. A natural solution is to add these conditioning variables c as an additional vector input to the neural network g and to parameterise β as another neural network. In this particular case, Equation (4.32) can be re-expressed as follows:

$$G(x|c;\theta) := \int_0^x g(t,c;\theta_g)dt + \beta(c;\theta_\beta) \ , \tag{4.33}$$

where the parameters of the monotonic transformation are $\theta = \theta_g \cup \theta_\beta$. Evaluating the function G requires solving an integral. This operation is performed numerically via the Clenshaw-Curtis quadrature, for the sake of efficiency.

In the scope of the distributional RL methodology at hand, the QF of the random return Z which takes as inputs quantile fractions $\tau \in [0,1]$ can be parameterised by a UMNN as $F_Z^{-1}(\tau|s,a;\theta) := G(\tau|s,a;\theta)$. Modelling the CDF of the random return Z requires the output to be bounded in $[0,1]$, which is achieved by passing the output of the UMNN through a sigmoid function σ: $F_Z(z|s,a;\theta) := \sigma(G(z|s,a;\theta))$. Modelling the random return PDF $p_Z(z|s,a;\theta)$ can be done via *normalizing flows* [78]. More precisely, it is achieved by using a

fixed latent distribution p_Y and exploiting the property that there exists a unique continuous monotonic function f satisfying the following equation (*change of variables theorem*) [75]:

$$p_Z(z|s,a;\theta) = p_Y(f(z|s,a;\theta))\left|\frac{\partial f}{\partial z}\right| . \tag{4.34}$$

The representation of p_Z is achieved by modelling the function f with a UMNN and fixing p_Y to an isotropic normal distribution. With such a representation, drawing samples from p_Z is performed by drawing samples from p_Y and applying the function f^{-1}. This requires inverting the UMNN, which can be done numerically by using any inversion method such as a binary search, since the inverse of a monotonic function is also monotonic.

Implementation details about UMNN for distributional RL

As previously indicated, the UMNN requires the solving of an integral, which may be a computationally expensive operation. For the sake of efficiency, this integral is numerically computed via the Clenshaw-Curtis quadrature. This technique presents the key advantage of converging exponentially fast for Lipshitz functions. In practice, a few function evaluations are required for reaching satisfying accuracy, and these operations present the advantage of being executable in parallel. This makes the complete forward computation of the UMNN quite efficient. Regarding the backward pass, the Leibniz rule can be used to make it more memory efficient. This technique enables to compute the derivative of an integral with respect to its inputs as the integral of the derivatives. For the interested reader, more details about the complete implementation of both forward and backward computations can be found in Appendix B of the research paper originally introducing the UMNN architecture [75].

Another key operation which has to be efficiently implemented is the computation of the expectation of the random return Z^π. Indeed, this important quantity will be repeatedly evaluated in distributional RL, since the decision-making policy π selects the action maximising the expectation of the random return. The methodology employed for that purpose is described hereafter for the three different probability distribution representations considered.

PDF representation. The PDF of the random return Z^π is modelled with a UMNN as $p_{Z^\pi}(z) = g(z)\sigma'\left(\int_0^z g(t)dt + \beta\right)$, where the function $\sigma'(\cdot)$ denotes the PDF of a normal distribution (or equivalently the derivative of a sigmoid function). Consequently, the expectation of the random return Z^π can be expressed as follows:

$$\mathbb{E}\left[Z^\pi\right] = \int_{z_{\min}}^{z_{\max}} z g(z)\sigma'\left(\int_0^z g(t)dt + \beta\right) dz . \tag{4.35}$$

A straightforward but inefficient solution would be to independently solve each inner integral for different values of the return z. Instead, for improved efficiency, these inner integrals are solved simultaneously by making use of the same neural network evaluation multiple times. The UMNN is first evaluated at evenly separated points between $z_{\min}$ and $z_{\max}$, and a composite Simpson's rule is then applied to approximate the inner integrals. Afterwards, the expectation of the random return Z^π is finally computed by estimating the outer integral using a *Monte Carlo* approach.

CDF representation. The CDF of the random return Z^π can be modelled with a UMNN as $F_{Z^\pi}(z) = \sigma\left(\int_0^z g(t)dt + \beta\right)$, where the function $\sigma(\cdot)$ is a sigmoid function (or equivalently the CDF of a normal distribution). Consequently, the PDF of the random return can be directly derived as $p_{Z^\pi}(z) = g(z)\sigma'\left(\int_0^z g(t)dt + \beta\right)$, and the expectation of the random return Z^π can be evaluated by following the methodology described in the previous paragraph.

QF representation. When the random return Z^π probability distribution is represented through its QF, no particular improvement is implemented and the expectation is estimated using *Monte Carlo*, similarly to the state-of-the-art QR-DQN, IQN and FQF algorithms.

4.4.3 Unconstrained monotonic deep Q-network algorithm

This section presents the *unconstrained monotonic deep Q-network* (UMDQN) algorithm, a novel generic distributional RL algorithm based on the methodology previously introduced in Section 4.4.1 and taking advantage of the UMNN architecture for representing the continuous probability distribution of the random return. More precisely, this research work details three versions of the generic UMDQN distributional RL algorithm: the UMDQN-KL, UMDQN-C and UMDQN-W algorithms, which respectively learn the continuous PDF, CDF and QF of the random return Z^π by minimising the KL divergence, Cramer distance and Wasserstein distance. Consequently, in contrast to previous works on distributional RL, the proposed approach presents the key advantage of offering a choice regarding the representation of the probability distribution together with the probability metric to work with. An illustration of these three novel distributional RL algorithms is provided in Figure 4.5.

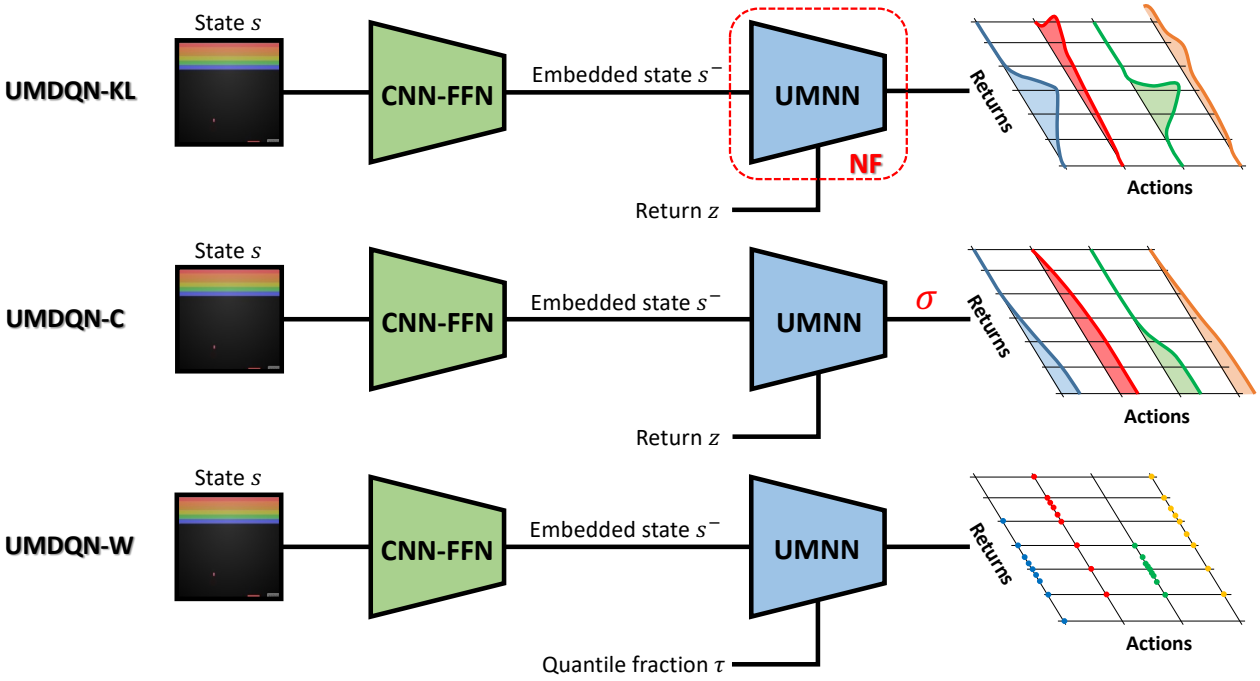


Figure 4.5: Illustration of the three versions of the UMDQN algorithm in the context of Atari games.

The UMDQN algorithm is an *off-policy* and *value iteration* DRL algorithm which is based on the same procedure as the popular DQN algorithm for generating trajectories and learning from that information. Numerous experiences $e = (s, a, r, s')$ are generated by sequentially interacting with the environment and are stored into an *experience replay memory* of fixed size with a first-in-first-out (FIFO) replacement policy. Additionally, a *target network*, whose parameters are denoted θ^-, is used for fixing the Bellman probability distribution to be learnt and is updated at regular intervals. As far as exploration is concerned, it is ensured through the use of the ϵ-greedy technique. At regular intervals during the interaction between the agent and its environment, batches of experiences are sampled from the replay memory to compute *Monte Carlo* estimates of an approximation of the loss defined in Equation (4.10) and perform stochastic gradient descent.

In fact, three important approximations are made regarding the loss previously defined in Equation (4.10). The first one results from the evaluation of the loss in expectation over the distribution of state-action pairs *sampled* from the environment. The second approximation originates from the computation of the expectation $\mathbb{E}_{s',r}$ in Equations (4.12), (4.14), (4.16) *outside* the probability metric $\mathcal{L}$. The last approximation comes from the estimation of the two expectations $\mathbb{E}_{s,a}$ and $\mathbb{E}_{s',r}$ using *Monte Carlo* with the experiences sampled from the replay memory. The second approximation may potentially introduce a bias, as it has already been demonstrated for the Wasserstein distance [73]. However, there is a solution for this probability metric in particular: the *(conditional) quantile regression* method [88]. This approach is claimed to allow for the unbiased stochastic approximation of the QF, and is adopted in the QR-DQN, IQN and FQF algorithms.

The learning process of the UMDQN algorithm is rigorously described in Algorithm 3. Within this description, $G_Z(\cdot|s, a; \theta)$ denotes the random return probability distribution modelled by a UMNN with parameters θ for the state-action pair (s, a), the operator T^π is defined in Equation (4.36) and reproduces the effect of the distributional Bellman operator on $G_Z(\cdot|s, a; \theta)$ in line with Equations (4.12), (4.14) and (4.16), the function L computes the error according to the probability metric selected, and $\mathcal{X}$ is a discretisation of the domain of the function representing probability distribution of the random return. The policy π selects the action maximising the expectation of the random return Z^π learnt so far.

$$T^\pi G_Z(x|s, a; \theta) = \begin{cases} \frac{1}{\gamma} G_Z\left(\frac{x-r}{\gamma}\middle| s', \pi(s'); \theta\right) & \text{if the UMNN models a PDF,} \\ G_Z\left(\frac{x-r}{\gamma}\middle| s', \pi(s'); \theta\right) & \text{if the UMNN models a CDF,} \\ r + \gamma G_Z(x|s', \pi(s'); \theta) & \text{if the UMNN models a QF.} \end{cases} \tag{4.36}$$

Algorithm 3 Learning process of the UMDQN algorithm

Sample a batch of N_e experiences $e = (s, a, r, s')$ from the replay memory.
Determine for each experience the next optimal action $a' = \pi(s') = \text{argmax}_{a \in \mathcal{A}} \mathbb{E}\left[G_Z(s', a; \theta^-)\right]$.
Compute the loss $\hat{\mathcal{L}} = \frac{1}{N_e} \sum_{s,a,r,s'} \left[\sum_{x \in \mathcal{X}} \left[L\left(T^\pi G_Z(x|s, a; \theta^-),\ G_Z(x|s, a; \theta)\right)\right]\right]$.
Optimise the UMNN parameters θ according to the resulting gradients $\nabla \hat{\mathcal{L}}$.

Implementation details about the UMDQN algorithm

To end this important section about the novel distributional RL algorithm introduced, some implementation details are shared together with the detailed pseudocodes for the three versions of the UMDQN algorithm presented. More precisely, the UMDQN-KL, UMDQN-C and UMDQN-W are thoroughly explained in Algorithms 4, 5 and 6, respectively.

Modelling the probability distribution of the random return for a terminal state may be tricky and deserves a brief discussion. In this case, the RL agent shall not receive any future rewards, and the random return distribution degenerates into a Dirac distribution shifted by the value of the last reward. In practice, such a particular probability distribution may be quite difficult to approximate with a DNN, depending on the distribution representation (PDF, CDF, QF). Additionally, this may potentially lead to numerical instabilities when computing the loss. For these reasons, this research work makes the choice to smooth out the Dirac distribution whenever appropriate. For the UMDQN-KL algorithm learning a PDF, a normal distribution with a tiny standard deviation is used as a replacement for the problematic Dirac distribution. For the UMDQN-C algorithm which is based on the random return CDF, the step function with infinite slope is supplanted by a smoother version with a large constant slope. Finally, the case of the UMDQN algorithm is left untouched since the QF of a Dirac distribution is trivial to model with a DNN (constant function).

As previously explained in this section, the loss defined in Equation (4.10) is approximated in the UMDQN algorithm, which may introduce a bias. This problem has already been demonstrated for the Wasserstein distance [73] and a solution has been proposed [81]: the *(conditional) quantile regression* method [88]. Without going into too much detail, this alternative approach is based on the *quantile regression loss*, which is an asymmetric convex loss function respectively penalising overestimation and underestimation errors with weights τ and $1-\tau$, with $\tau \in [0,1]$ being a quantile fraction. The UMDQN-W algorithm takes advantage of this methodology, similarly to the state-of-the-art QR-DQN, IQN and FQF distributional RL algorithms. In fact, to ensure smoothness at zero, a slightly modified quantile regression loss is used by these algorithms, the *quantile Huber loss* which is defined for the error $x \in \mathbb{R}$ as the following:

$$\rho_\tau^\kappa(x) = \left|\tau - 1_{\{x<0\}}\right| \frac{\mathcal{H}_\kappa(x)}{\kappa} \ , \tag{4.37}$$

$$\mathcal{H}_\kappa(x) = \begin{cases} \frac{1}{2}x^2 & \text{if } |x| \leq \kappa, \\ \kappa(|x| - \frac{1}{2}\kappa) & \text{otherwise,} \end{cases} \tag{4.38}$$

where the threshold κ is a parameter to be tuned. An illustration of the quantile Huber loss with $\kappa = 1$ is provided in Figure 4.6 below. This alternative loss function is evaluated on the pairwise temporal difference (TD) errors δ_{ij} expressed as follows:

$$\delta_{ij} = r + \gamma F_{Z^\pi}^{-1}\left(\tau_j | s', \pi(s')\right) - F_{Z^\pi}^{-1}\left(\tau_i | s, a\right) \ . \tag{4.39}$$

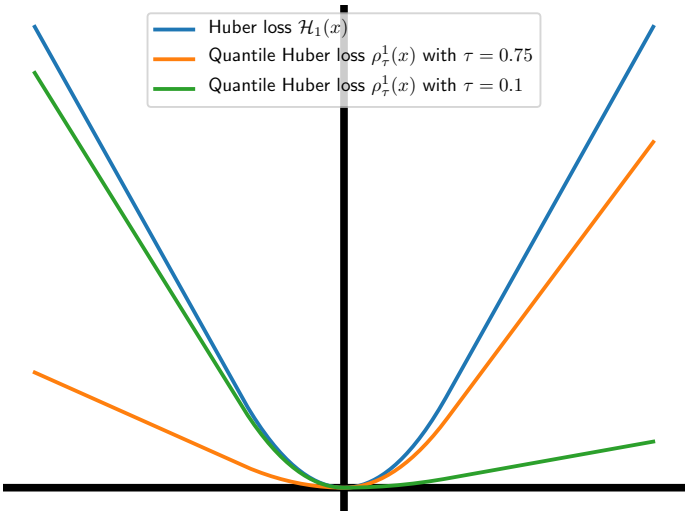


Figure 4.6: Illustration of the quantile Huber loss.

Algorithm 4 UMDQN-KL algorithm

Initialise the experience replay memory M of capacity C.
Initialise the main UMNN weights θ (Xavier initialisation).
Initialise the target UMNN weights $\theta^- = \theta$.
for episode $= 0$ **to** N **do**
  for $t = 0$ **to** T, or until episode termination **do**
    Acquire the state s from the environment $\mathcal{E}$.
    With probability ϵ, select a random action $a \in \mathcal{A}$.
    Otherwise, select $a = \text{argmax}_{a' \in \mathcal{A}} \mathbb{E}\left[G_Z(s, a'; \theta)\right]$.
    Interact with the environment $\mathcal{E}$ with action a to get the next state s' and the reward r.
    Store the experience $e = (s, a, r, s')$ in M.
    if $t \% T' = 0$ **then**
      Randomly sample from M a minibatch of N_e experiences $e_i = (s_i, a_i, r_i, s_i^{'})$.
      Derive a discretisation of the domain $\mathcal{X}$ by sampling N_z returns $z \sim \mathcal{U}([z_{\min}, z_{\max}])$.
      for $i = 0$ **to** N_e **do**
        for all $z \in \mathcal{X}$ **do**
          if $s_i^{'}$ is terminal **then**
            Set $y_i(z) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{z-\mu}{\sigma}\right)^2\right)$ with $\mu = r_i$ and $\sigma = \frac{z_{\max} - z_{\min}}{N_z}$.
          else
            Set $y_i(z) = \frac{1}{\gamma}\, G_Z\left(\frac{z - r_i}{\gamma} \middle| s_i^{'}, \text{argmax}_{a_i^{'} \in \mathcal{A}} \mathbb{E}\left[G_Z(s_i^{'}, a_i^{'}; \theta^-)\right]; \theta^-\right)$.
          end if
        end for
      end for
      Compute the loss $\mathcal{L}_{KL}(\theta) = \sum_{i=0}^{N_e}\left(\sum_{z \in \mathcal{X}}\ y_i(z) \log\left(\frac{y_i(z)}{G_Z(z|s_i, a_i; \theta)}\right)\right)$.
      Clip the resulting gradient in the range $[0, 1]$.
      Update the main UMNN parameters θ using the ADAM optimiser.
    end if
    Update the target UMNN parameters $\theta^- = \theta$ every N^- steps.
    Anneal the ϵ-greedy exploration parameter ϵ.
  end for
end for

Algorithm 5 UMDQN-C algorithm

Initialise the experience replay memory M of capacity C.
Initialise the main UMNN weights θ (Xavier initialisation).
Initialise the target UMNN weights $\theta^- = \theta$.
for episode $= 0$ **to** N **do**
  for $t = 0$ **to** T, or until episode termination **do**
    Acquire the state s from the environment $\mathcal{E}$.
    With probability ϵ, select a random action $a \in \mathcal{A}$.
    Otherwise, select $a = \text{argmax}_{a' \in \mathcal{A}} \mathbb{E}\left[G_Z(s, a'; \theta)\right]$.
    Interact with the environment $\mathcal{E}$ with action a to get the next state s' and the reward r.
    Store the experience $e = (s, a, r, s')$ in M.
    if $t \% T' = 0$ **then**
      Randomly sample from M a minibatch of N_e experiences $e_i = (s_i, a_i, r_i, s_i^{'})$.
      Derive a discretisation of the domain $\mathcal{X}$ by sampling N_z returns $z \sim \mathcal{U}([z_{\min}, z_{\max}])$.
      for $i = 0$ **to** N_e **do**
        for all $z \in \mathcal{X}$ **do**
          if $s_i^{'}$ is terminal **then**
            Set $y_i(z) = \begin{cases} 0 & \text{if } z < r_i, \\ 1 & \text{otherwise.} \end{cases}$
          else
            Set $y_i(z) = G_Z\left(\frac{z - r_i}{\gamma} \middle| s_i^{'}, \text{argmax}_{a_i' \in \mathcal{A}} \mathbb{E}\left[G_Z(s_i^{'}, a_i^{'}; \theta^-)\right]; \theta^-\right)$.
          end if
        end for
      end for
      Compute the loss $\mathcal{L}_C(\theta) = \sum_{i=0}^{N_e} \left(\sum_{z \in \mathcal{X}} \left(y_i(z) - G_Z(z | s_i, a_i; \theta)\right)^2\right)^{1/2}$.
      Clip the resulting gradient in the range $[0, 1]$.
      Update the main UMNN parameters θ using the ADAM optimiser.
    end if
    Update the target UMNN parameters $\theta^- = \theta$ every N^- steps.
    Anneal the ϵ-greedy exploration parameter ϵ.
  end for
end for

Algorithm 6 UMDQN-W algorithm

Initialise the experience replay memory M of capacity C.
Initialise the main UMNN weights θ (Xavier initialisation).
Initialise the target UMNN weights $\theta^- = \theta$.
for episode $= 0$ **to** N **do**
  for $t = 0$ **to** T, or until episode termination **do**
    Acquire the state s from the environment $\mathcal{E}$.
    With probability ϵ, select a random action $a \in \mathcal{A}$.
    Otherwise, select $a = \text{argmax}_{a' \in \mathcal{A}} \mathbb{E}\left[G_Z(s, a'; \theta)\right]$.
    Interact with the environment $\mathcal{E}$ with action a to get the next state s' and the reward r.
    Store the experience $e = (s, a, r, s')$ in M.
    if $t \% T' = 0$ **then**
      Randomly sample from M a minibatch of N_e experiences $e_i = (s_i, a_i, r_i, s_i^{'})$.
      Sample N_τ values for the first quantile fraction $\tau_i \sim \mathcal{U}([0, 1])$.
      Sample N_τ values for the second quantile fraction $\tau_j \sim \mathcal{U}([0, 1])$.
      for $k = 0$ **to** N_e **do**
        for $i = 0$ **to** N_τ **do**
          for $j = 0$ **to** N_τ **do**

$$y_k(\tau_j) = \begin{cases} r_k & \text{if } s_k^{'} \text{ terminal,} \\ r_k + \gamma \; G_Z\left(\tau_j \middle| s_k^{'}, \text{argmax}_{a_k^{'} \in \mathcal{A}} \mathbb{E}\left[G_Z(s_k^{'}, a_k^{'}; \theta^-)\right]; \theta^-\right) & \text{otherwise.} \end{cases}$$

          end for
          $\delta_{ij}(k) = y_k(\tau_j) - G_Z(\tau_i | s_k, a_k; \theta)$.
        end for
      end for
      Compute the loss $\mathcal{L}_W(\theta) = \sum_{k=0}^{N_e} \left(\sum_{i=0}^{N_\tau} \mathbb{E}_j \left[\rho_{\tau_i}^{\kappa}(\delta_{ij}(k)) \right] \right)$.
      Clip the resulting gradient in the range $[0, 1]$.
      Update the main UMNN parameters θ using the ADAM optimiser.
    end if
    Update the target UMNN parameters $\theta^- = \theta$ every N^- steps.
    Anneal the ϵ-greedy exploration parameter ϵ.
  end for
end for

4.5 Results

4.5.1 Benchmark environments

The performance assessment methodology adopted by this research work to evaluate the performance of the UMDQN distributional RL algorithm includes four different types of benchmark environments, which are illustrated in Figures 4.7, 4.8 and 4.9:

- A stochastic grid world environment,
- A set of classic control environments,
- A set of Atari games,
- A set of MinAtar games.

The first benchmark environment is a *stochastic grid world* designed in the scope of this particular research on distributional RL. It consists of a 7×7 grid world, an environment which is commonly taken into consideration for analysing and evaluating the performance of RL algorithms. The objective of the agent is simply to reach a certain target location which is fixed, while avoiding a fixed trap. The particularity of this grid world is that both the transition and reward functions are stochastic (p_T and p_R), in order to better highlight the impact of the distributional RL approach. In addition to evaluating the policy performance, this specific environment will also be particularly useful for visualising and interpreting the random return probability distributions learnt by the distributional RL algorithm. More details about this benchmark environment can be found in the underlying MDP, which can be defined as the following:

- $\mathcal{S} \in \{0, ..., 6\} \times \{0, ..., 6\}$, a state s being composed of the two coordinates of the agent within the grid,
- $\mathcal{A} = \{\texttt{RIGHT}, \texttt{UP}, \texttt{LEFT}, \texttt{DOWN}\}$, with an action a being a moving direction,
- $p_R(r|s, a) \sim \mathcal{N}(\mu, \sigma^2)$ where:
  - $\mu = 1$ if the agent reaches the target location (terminal state),
  - $\mu = -1$ if the agent falls into the trap (terminal state),
  - $\mu = 0$ otherwise,
  - $\sigma = 0.1$ at anytime,
- $p_T(s'|s, a)$ associates a 50% chance to move twice in the chosen direction instead of once, while keeping the agent within the 7×7 grid world (no border crossing allowed),
- p_0 associates the exact same probability to all states $s_0 \in \mathcal{S}$, except for the two states corresponding to the trap and target locations which have a null probability,
- $\gamma = 0.5$.

The next type of benchmark environment is a set of four *classic control* problems from OpenAI Gym [89]: CartPole, Acrobot, MountainCar and LunarLander. Despite the general preference of the distributional RL community for Atari games over these simpler environments, the classic control problems remain particularly valuable and popular benchmarks for evaluating RL algorithms. Moreover, these environments are promoted by the article [90] which proposes an alternative set of benchmarks which are less computationally intensive. That particular work being quite interesting and well received by the RL research community, this thesis chapter adopts its suggestions. More details about the four classic control problems considered in this research work are provided hereafter:

- `CartPole-v0`: The objective is to balance a pole attached by a non-actuated joint to a cart moving along a frictionless track. The state is composed of four continuous values: the cart position, the cart velocity, the pole angle and the pole velocity at the tip. The agent's action is either to push the cart to the left or to the right. A reward of $+1$ is received for each time step with the pole remaining balanced. An episode terminates when the pole angle is more than $\pm 12°$ or when the cart reaches the edge of the display, but also if the episode length is greater than 200.

- `Acrobot-v1`: This system is composed of a double-jointed pendulum, with the joint between the two links being actuated. The objective is to swing the pendulum so that the end of the outer link reaches a given height. The state is a six-dimensional vector describing the system's angles and velocities. To achieve its goal, the agent has three actions at its disposal: either applying no torque, or applying a fixed torque to the left or to the right. The agent is given a reward of -1 for each time step before achieving the objective position. An episode either terminates when this objective is achieved or when the episode length exceeds 500.

- `MountainCar-v0`: The objective is to drive an underpowered car up a steep hill. To achieve that goal, the agent has to learn to leverage potential energy by driving back and forth for gaining momentum. The state consists of both the position and velocity of the car. The agent's action can either be to push the car to the left, do nothing or push the car to the right. A reward of -1 is received at each time step until the goal position is eventually reached. An episode terminates when this particular position is achieved, or if the episode length is greater than 200.

- `LunarLander-v2`: This environment consists of a simulated 2D world within which the objective is to safely land a lander with a limited amount of fuel on a target location. The RL state is composed of 8 values: the two coordinates of the lander, its linear velocities in the horizontal and vertical directions, its angle and angular velocity, as well as two booleans representing whether each leg is in contact with the ground or not. To achieve its objective, the agent has access to four actions: do nothing, fire the left orientation engine, fire the main engine and fire the right orientation engine. A reward between $+100$ and $+140$ is received for safely landing and coming to rest at the designated location. Additionally, a crash results in receiving a -100 reward, while coming to rest induces a reward of $+100$. There is also a $+10$ reward generated for each leg with ground contact. Finally, rewards of -0.3 and -0.03 are respectively obtained for

each time step firing the main and side engines. The termination of an episode occurs when the lander crashes or gets outside of the viewport, or when the lander is no longer awake (meaning that it does not move nor collide with any other body).

The third type of benchmark environment is a set of three *Atari games* from the Atari-57 benchmark [53]: Pong, Boxing and Freeway. The evaluation and analysis of distributional RL algorithms are generally performed on the complete Atari-57 benchmark, which offers a relevant performance assessment methodology but also presents some important drawbacks for distributional RL. Indeed, these environments are mostly deterministic and require a tremendous amount of computational power. Since the original publication of the Atari-57 benchmark, diverse evaluation methodologies have progressively appeared. In this research work, the best practices proposed by the article [91] are adopted. Moreover, the mostly deterministic transitions within Atari games are made stochastic by using the sticky action generalisation technique (stochastic transitions, but still deterministic rewards). This last addition makes the Atari environments considered slightly more complex compared to the ones from previous publications in distributional RL. In practice, the implementation adopted is the `{}NoFrameskip` from OpenAI gym [89], together with the following wrappers:

- Formatting of a frame to 84×84 pixels,
- Normalisation of the values of the pixels,
- Clipping of the reward to $\{+1,\ 0,\ -1\}$,
- Sending of the episode termination signal when all the agent's lives are lost,
- Execution of a random number of `NOOP` actions at the beginning of an episode (max 30),
- Execution of sticky actions with a 0.25 probability,
- Frame skipping and maximisation operation with period 4,
- Stacking of the final 4 frames.

The last type of benchmark environment is a set of five *MinAtar games* [92]: Asterix, Breakout, Freeway, Seaquest and SpaceInvaders. These environments are miniaturised and slightly simplified versions of five Atari games representative of the complete Atari-57 suite. The objective behind these MinAtar environments is to make RL experimentation around Atari games much more accessible and efficient. To do so, MinAtar reduces the representation complexity of five representative Atari games, while avoiding as much as possible altering the mechanics of the original games. The alternative state representation is of dimension $10 \times 10 \times n$ and binary, where n is the number of channels representing a game-specific object. Moreover, MinAtar games include stochasticity in the form of sticky actions and randomised spawn locations, which is particularly important for analysing distributional RL algorithms. Finally, this alternative benchmark is also promoted by the same article [90] as a replacement for the Atari-57 benchmark in order to achieve more inclusive DRL research.

To end this section about the performance assessment methodology, an argument for bypassing the full Atari-57 benchmark generally adopted in research works about distributional RL is presented. As previously hinted, the computational cost associated with this particular benchmark is significant. In this case, two entire weeks' worth of computations are required for training one RL agent on a single Atari game using the UMDQN algorithm with hardware acceleration enabled (NVIDIA RTX 2080 Ti). Therefore, running this novel distributional RL algorithm for five different random seeds on the full Atari-57 benchmark would approximately require $57 * 5 * 14 \simeq 4000$ days when having access to a single GPU, without even considering the hyperparameters tuning phase. It naturally becomes totally impracticable without parallelisation with numerous GPUs. Although the UMDQN algorithm presents the drawback of being slightly more computationally expensive compared to the state-of-the-art distributional RL algorithms, the previous conclusion remains in line with findings from the scientific literature. For instance, the simpler DQN algorithm takes roughly 1425 days to fully train for all Atari games using specialised hardware (NVIDIA Tesla P100) [90]. Because this problem creates a real barrier to entry for modest laboratories having access to a limited amount of computational power, it is repeatedly discussed by the RL research community. For this reason, the present research work adopts a different yet insightful set of benchmark environments for evaluating distributional RL algorithms, for more inclusive DRL research.

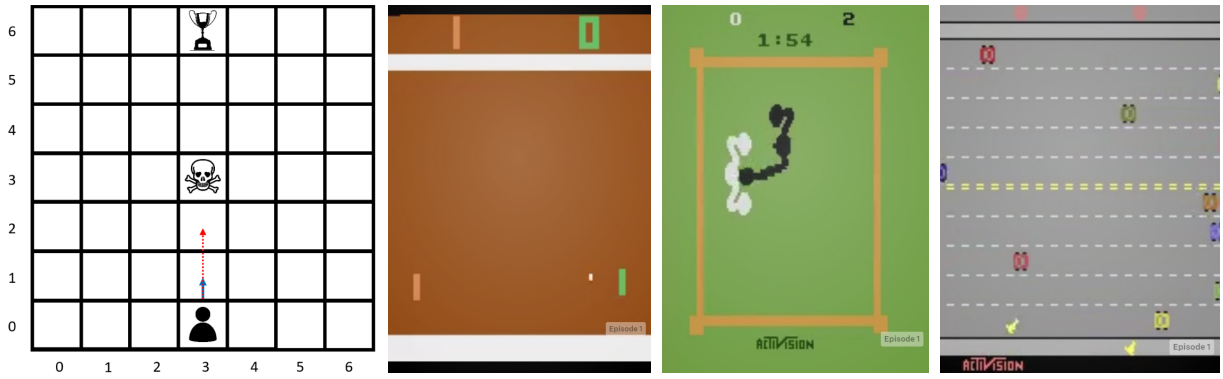


Figure 4.7: Illustration of some benchmark environments with, from left to right, the stochastic grid world and the Atari games Pong, Boxing and Freeway.

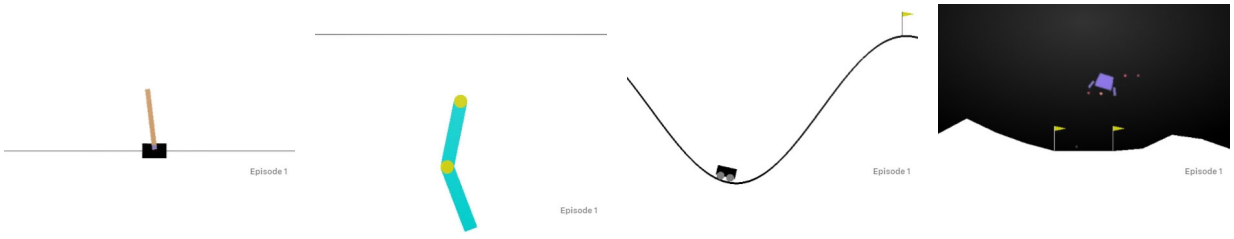


Figure 4.8: Illustration of some benchmark environments with, from left to right, the CartPole, Acrobot, MountainCar and LunarLander classic control problems.

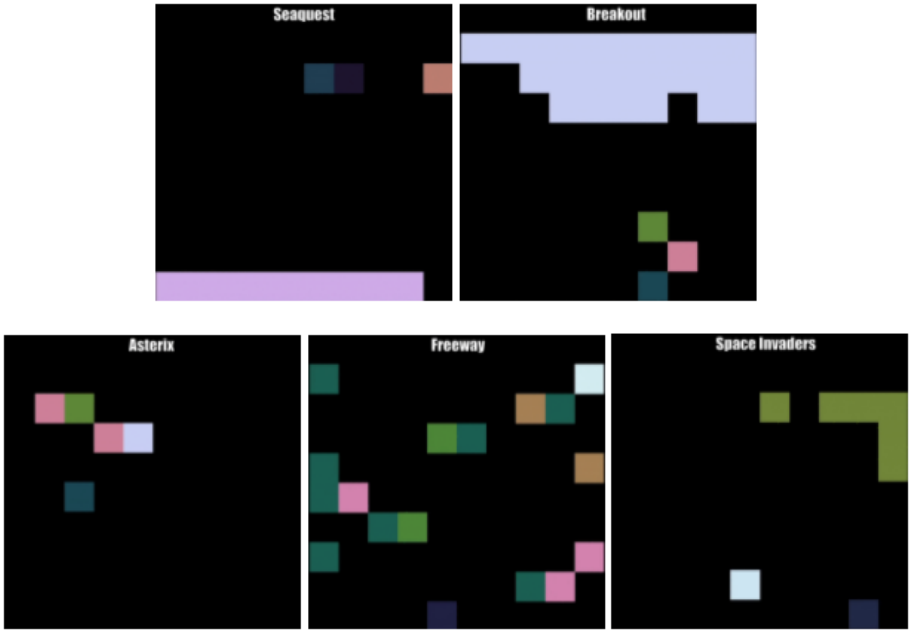


Figure 4.9: Illustration of some benchmark environments with all the MinAtar games.

4.5.2 Experiments reproducibility

In order to ensure the reproducibility of the experiments presented in this research work, this section provides the main hyperparameters adopted. Additionally, the complete code used for generating the results discussed in the next section is made publicly available at the following link:

`https://github.com/ThibautTheate/Unconstrained-Monotonic-Deep-Q-Network-algorithm`.

The most important criterion taken into account for the selection/tuning of these hyperparameters is the fair comparison between the DRL algorithms studied, while considering at the same time the values reported by the state-of-the-art distributional RL algorithms. First of all, Table 4.3 provides a brief description of the hyperparameters to be tuned in the scope of this research work. Then, Table 4.4 presents the domain $\mathcal{X}$ selected for the different benchmark environments studied (lower and upper bounds). Finally, Tables 4.5, 4.6, 4.7 and 4.8 present the hyperparameters used for generating the results for all the benchmark environments previously presented in Section 4.5.1.

A complicated choice when it comes to hyperparameters tuning concerns the domain $\mathcal{X}$. Firstly, is it fairer to have the same lower bound $z_{\min}$ and upper bound $z_{\max}$ for the domain $\mathcal{X}$ for all the benchmark environments as it is generally done in scientific literature, or slightly tune these two hyperparameters for each environment? Secondly, if tuned, how to efficiently select relevant values for these two bounds without requiring complicated analyses? This research work makes the choice to specialise the domain $\mathcal{X}$ for each benchmark environment.

This decision is motivated by the diversity of the environments, with completely different ranges for the random return. For each control problem, a short analysis is performed to estimate the minimum and maximum returns based on the shape of the reward probability distribution p_R.

Let's consider for instance the CartPole environment. Knowing that a $+1$ reward is obtained at each time step until episode termination and that the discount factor is equal to 0.99, it is possible to estimate the lower and upper bounds for the returns. In the worst case, the agent totally fails and the episode ends after a few time steps, meaning that the minimum return is close to 0. On the contrary, if the agent manages to continuously keep the pole balanced, the maximum return can be estimated as the following:

$$z_{\max} = \sum_{i=0}^{\infty} 0.99^i \simeq 100 \ .$$

After consideration of a small margin, an appropriate domain $\mathcal{X}$ is obtained for this particular control problem. The same analysis can be repeated for the other benchmark environments, resulting in Table 4.4.

Table 4.3: Description of the main hyperparameters associated with the distributional RL algorithms studied.

Hyperparameter	**Description**
Network structure	Structure of the DNN representing the random return Z^π (neurons per layer).
Discount factor	Discount factor γ adopted for the Q-learning update.
Learning rate	Learning rate of the DL optimiser (ADAM).
Optimiser epsilon	Epsilon of the DL optimiser (ADAM) to improve numerical stability.
Main update frequency	Frequency T' (in number of steps) at which the main network is updated.
Target update frequency	Frequency N^- (in number of steps) at which the target network is updated.
Replay memory capacity	Capacity C (in number of experiences) of the experience replay memory M.
Batch size	Size of the batch N_e (in experiences) used for each gradient descent iteration.
ϵ-greedy start	Initial value of ϵ, for the ϵ-greedy exploration technique.
ϵ-greedy end	Final value of ϵ, for the ϵ-greedy exploration technique.
ϵ-greedy decay	Exponential decay (in steps) of ϵ, for the ϵ-greedy exploration technique.
ϵ-greedy test	Value of ϵ when testing the policy, for the ϵ-greedy exploration technique.
Number of z values	Number of returns N_z used for representing distributions (PDF and CDF).
Number of τ values	Number of quantile fractions N_τ used for representing distributions (QF).

Table 4.4: Domain $\mathcal{X}$ set for the different benchmark environments.

Benchmark environment	**Lower bound of $\mathcal{X}$**	**Upper bound of $\mathcal{X}$**
Stochastic grid world	-2	2
Atari Pong	-5	5
Atari Boxing	-1	10
Atari Freeway	-1	10
CartPole	-10	110
Acrobot	-110	10
MountainCar	-110	10
LunarLander	-150	200
MinAtar Asterix	-1	10
MinAtar Breakout	-1	10
MinAtar Freeway	-1	10
MinAtar Seaquest	-1	10
MinAtar SpaceInvaders	-1	20

Table 4.5: Hyperparameters selected for the stochastic grid world benchmark environment.

Hyperparameter	**UMDQN-KL**	**UMDQN-C**	**UMDQN-W**
Network structure	$[128]_{\text{DNN}} + [128]_{\text{UMNN}}$	$[128]_{\text{DNN}} + [128]_{\text{UMNN}}$	$[128]_{\text{DNN}} + [128]_{\text{UMNN}}$
Discount factor	0.5	0.5	0.5
Learning rate	10^{-4}	10^{-4}	10^{-4}
Optimiser epsilon	10^{-5}	10^{-5}	10^{-5}
Main update frequency	1	1	1
Target update frequency	1000	1000	1000
Replay memory capacity	10^4	10^4	10^4
Batch size	32	32	32
ϵ-greedy start	1.0	1.0	1.0
ϵ-greedy end	0.01	0.01	0.01
ϵ-greedy decay	10^4	10^4	10^4
ϵ-greedy test	0.001	0.001	0.001
Number of z values	200	200	-
Number of τ values	-	-	200

Table 4.6: Hyperparameters selected for the Atari games benchmark environments.

Hyperparameter	**UMDQN-KL**	**UMDQN-C**	**UMDQN-W**
Network structure	DQN + $[128]_{\text{UMNN}}$	DQN + $[128]_{\text{UMNN}}$	DQN + $[128]_{\text{UMNN}}$
Discount factor	0.99	0.99	0.99
Learning rate	5×10^{-5}	5×10^{-5}	5×10^{-5}
Optimiser epsilon	10^{-5}	10^{-5}	10^{-5}
Main update frequency	4	4	4
Target update frequency	10^4	10^4	10^4
Replay memory capacity	10^5	10^5	10^5
Batch size	32	32	32
ϵ-greedy start	1.0	1.0	1.0
ϵ-greedy end	0.01	0.01	0.01
ϵ-greedy decay	10^6	10^6	10^6
ϵ-greedy test	0.001	0.001	0.001
Number of z values	200	200	-
Number of τ values	-	-	200

Table 4.7: Hyperparameters selected for the classic control benchmark environments.

Hyperparameter	**UMDQN-KL**	**UMDQN-C**	**UMDQN-W**
Network structure	$[128]_{\text{DNN}} + [128]_{\text{UMNN}}$	$[128]_{\text{DNN}} + [128]_{\text{UMNN}}$	$[128]_{\text{DNN}} + [128]_{\text{UMNN}}$
Discount factor	0.99	0.99	0.99
Learning rate	10^{-4}	10^{-4}	10^{-4}
Optimiser epsilon	10^{-5}	10^{-5}	10^{-5}
Main update frequency	1	1	1
Target update frequency	1000	1000	1000
Replay memory capacity	10^4	10^4	10^4
Batch size	32	32	32
ϵ-greedy start	1.0	1.0	1.0
ϵ-greedy end	0.01	0.01	0.01
ϵ-greedy decay	10^4	10^4	10^4
ϵ-greedy test	0.001	0.001	0.001
Number of z values	200	200	-
Number of τ values	-	-	200

Table 4.8: Hyperparameters selected for the MinAtar games benchmark environments.

Hyperparameter	**UMDQN-KL**	**UMDQN-C**	**UMDQN-W**
Network structure	DQN + $[128]_{\text{UMNN}}$	DQN + $[128]_{\text{UMNN}}$	DQN + $[128]_{\text{UMNN}}$
Discount factor	0.99	0.99	0.99
Learning rate	5×10^{-5}	5×10^{-5}	5×10^{-5}
Optimiser epsilon	10^{-5}	10^{-5}	10^{-5}
Main update frequency	4	4	4
Target update frequency	10^4	10^4	10^4
Replay memory capacity	10^5	10^5	10^5
Batch size	32	32	32
ϵ-greedy start	1.0	1.0	1.0
ϵ-greedy end	0.01	0.01	0.01
ϵ-greedy decay	10^6	10^6	10^6
ϵ-greedy test	0.001	0.001	0.001
Number of z values	200	200	-
Number of τ values	-	-	200

4.5.3 Results discussion

Probability distribution analysis. Besides the evaluation of the resulting policy performance, it is important to rigorously assess the correctness of the probability distributions learnt by a distributional RL algorithm. In fact, such a valuable investigation is generally lacking in the scientific literature. As previously hinted, this particular analysis is performed on the stochastic grid world environment. Since the underlying control problem is relatively easy to solve from a human perspective, an optimal policy π^* can be manually derived without any algorithm for that environment. This property significantly eases the assessment of the soundness of the random return probability distributions learnt. Once the optimal policy is available, the true probability distributions of the random return can be effectively estimated via Monte Carlo. The same operation should also be performed with the policy learnt by the distributional RL algorithm, since incorrect probability distributions could also be caused by suboptimal policies. Based on this methodology, Figure 4.10 graphically compares the probability distributions learnt by the three versions of the UMDQN algorithm (PDF, CDF and QF) with the true random return distributions associated with an optimal policy for a particular state of the environment. Although the PDF and CDF respectively learnt by the UMDQN-KL and UMDQN-C algorithms are not entirely accurate, they remain qualitatively very similar to the true random return distribution, with among others the multimodality preserved (see blue line). This key observation not only validates the soundness of the probability distributions learnt, but also confirms that these two distributional RL algorithms can effectively learn an optimal policy for this benchmark environment. Conversely, the error made by the UMDQN-W algorithm learning the QF of the random return is much more concerning. In this case, it is clear that the distribution multimodality is no longer preserved for instance (see blue line). Additional analyses reveal that this difference does not originate from a suboptimal policy learnt by the distributional RL algorithm. In fact, this critical

observation is consistent with Equations (4.15) and (4.16) together with the explanation provided in Section 4.4.1 regarding the learning of the QF on the basis of the distributional Bellman operator. The expectation of the random return is preserved, but the probability distribution higher-order moments are not. As previously explained, this problem is not specific to the UMDQN-W algorithm and applies to several state-of-the-art distributional RL algorithms. To illustrate that claim, Figure 4.10 plots the probability distributions of the random return learnt by the CDQN, QR-DQN, IQN and FQF algorithms, that do all achieve an optimal policy for the stochastic grid world environment. On the one hand, the CDQN algorithm learning the categorical PDF of the random return based on the KL divergence achieves satisfying results, which is in line with the previous observation for the UMDQN-KL algorithm. On the other hand, the QR-DQN, IQN and FQF algorithms clearly show their limitations for accurately modelling the QF of the random return. Therefore, this particular learning methodology adopted by several state-of-the-art distributional RL algorithms should only be considered when the objective is to learn policies maximising the expectation of the random return, but should instead be discarded when the intention is to exploit the complete probability distribution, for learning risk-sensitive policies for instance.

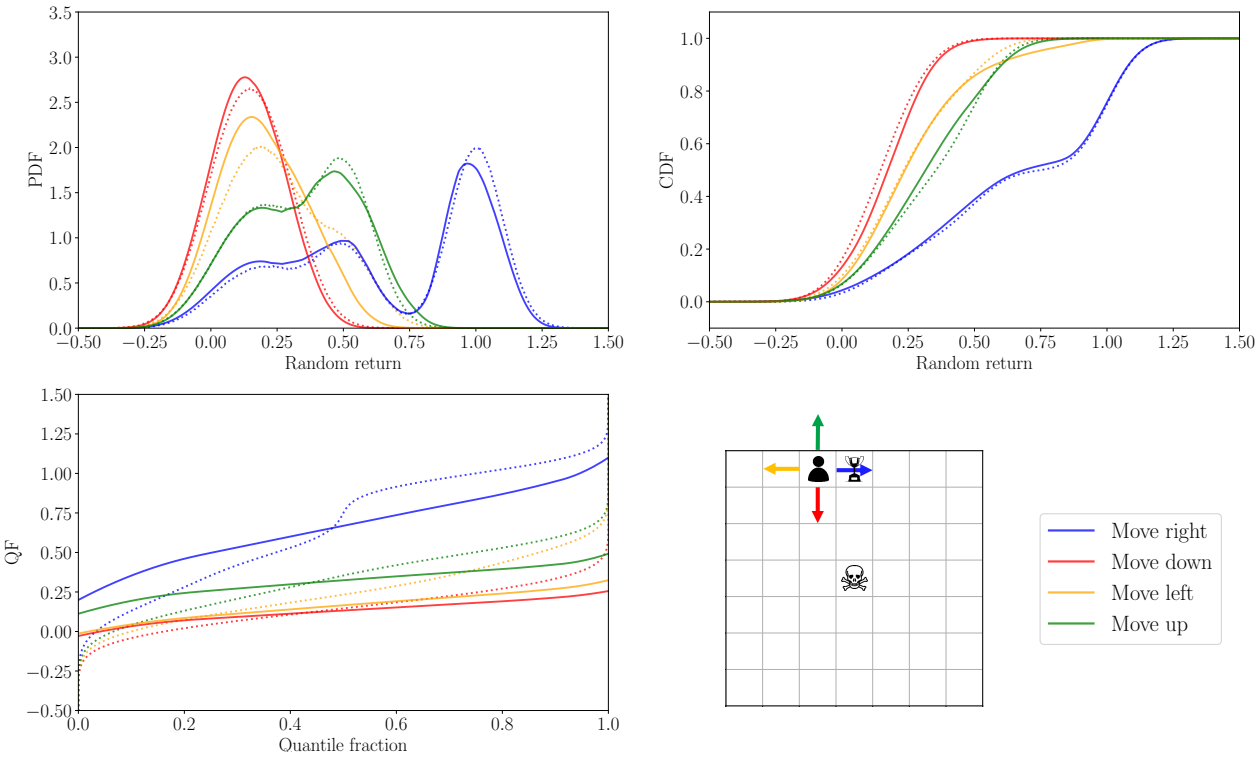


Figure 4.10: Comparison of the random return probability distributions (PDF, CDF and QF) respectively learnt by the UMDQN-KL, UMDQN-C and UMDQN-W algorithms (plain lines) with the true random return probability distributions (PDF, CDF and QF) estimated via Monte Carlo and associated with an optimal policy (dotted lines), for a particular state of the stochastic grid world environment.

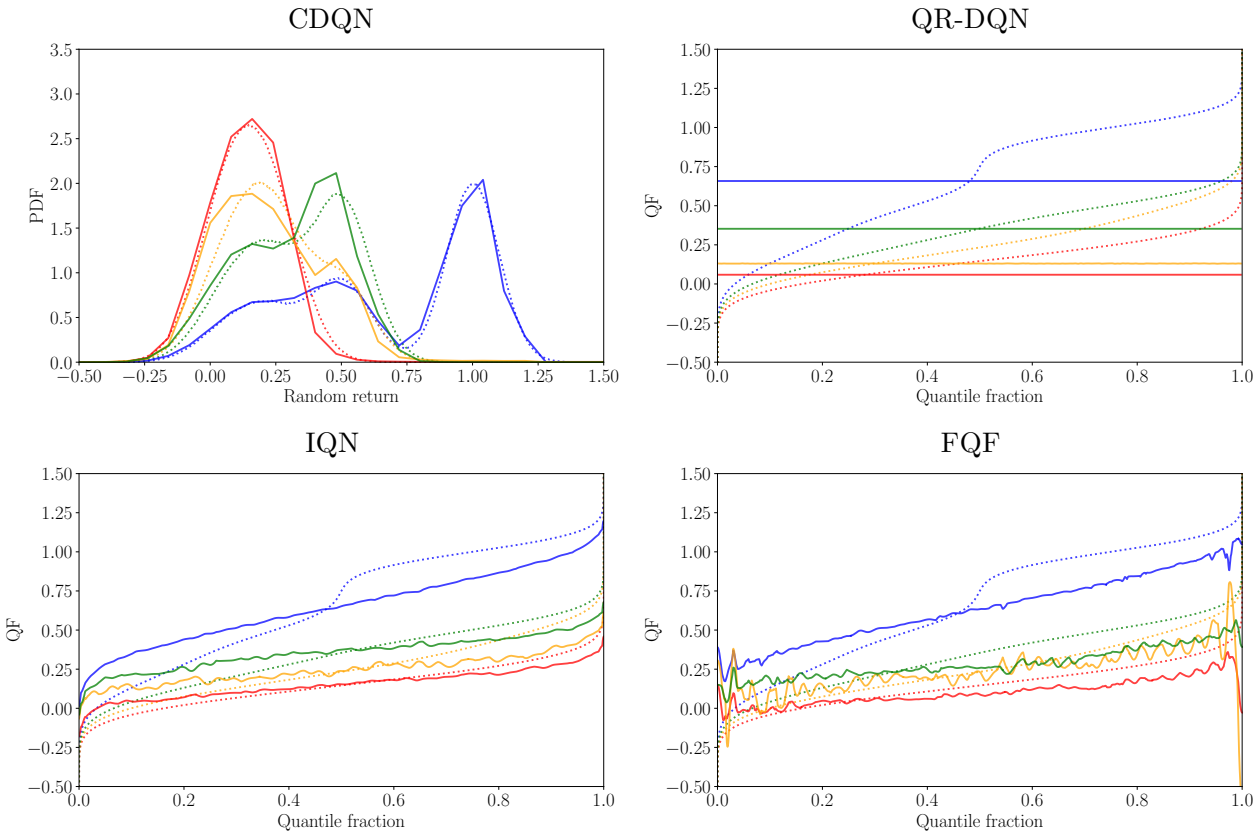


Figure 4.11: Comparison of the random return probability distributions learnt by the CDQN, QR-DQN, IQN and FQF state-of-the-art algorithms (plain lines) with the true random return probability distributions estimated via Monte Carlo and associated with an optimal policy (dotted lines), for a particular state of the stochastic grid world environment.

Policy performance. As far as the quality of the decision-making policies π learnt by the distributional RL algorithms is concerned, Figure 4.12 presents the results achieved by the three versions of the UMDQN algorithm for all the benchmark environments introduced in Section 4.5.1. The policy performance plotted is the cumulative reward achieved by the RL agent over one full episode. For the sake of reliability, the results are averaged over five different random seeds and the variance is highlighted. Moreover, for improved readability, a moving average operation is performed to further smooth the curves. Taking into account their respective strengths and weaknesses detailed hereafter, it is quite difficult to identify a clear winner overall in terms of policy performance among the three versions of the UMDQN algorithm, even though the UMDQN-KL algorithm certainly lags behind the other two. Since the same function approximator class is used, this conclusion also stands for the distribution representations and the probability metrics underneath the distributional RL algorithms. An argument for potentially explaining this observation is the fact that a neural network may more efficiently model the PDF, CDF or QF of the random return distributions depending on the characteristics of these particular probability distributions, such as the multimodality or the values of the moments. Another hypothesis is to point out the approximation of

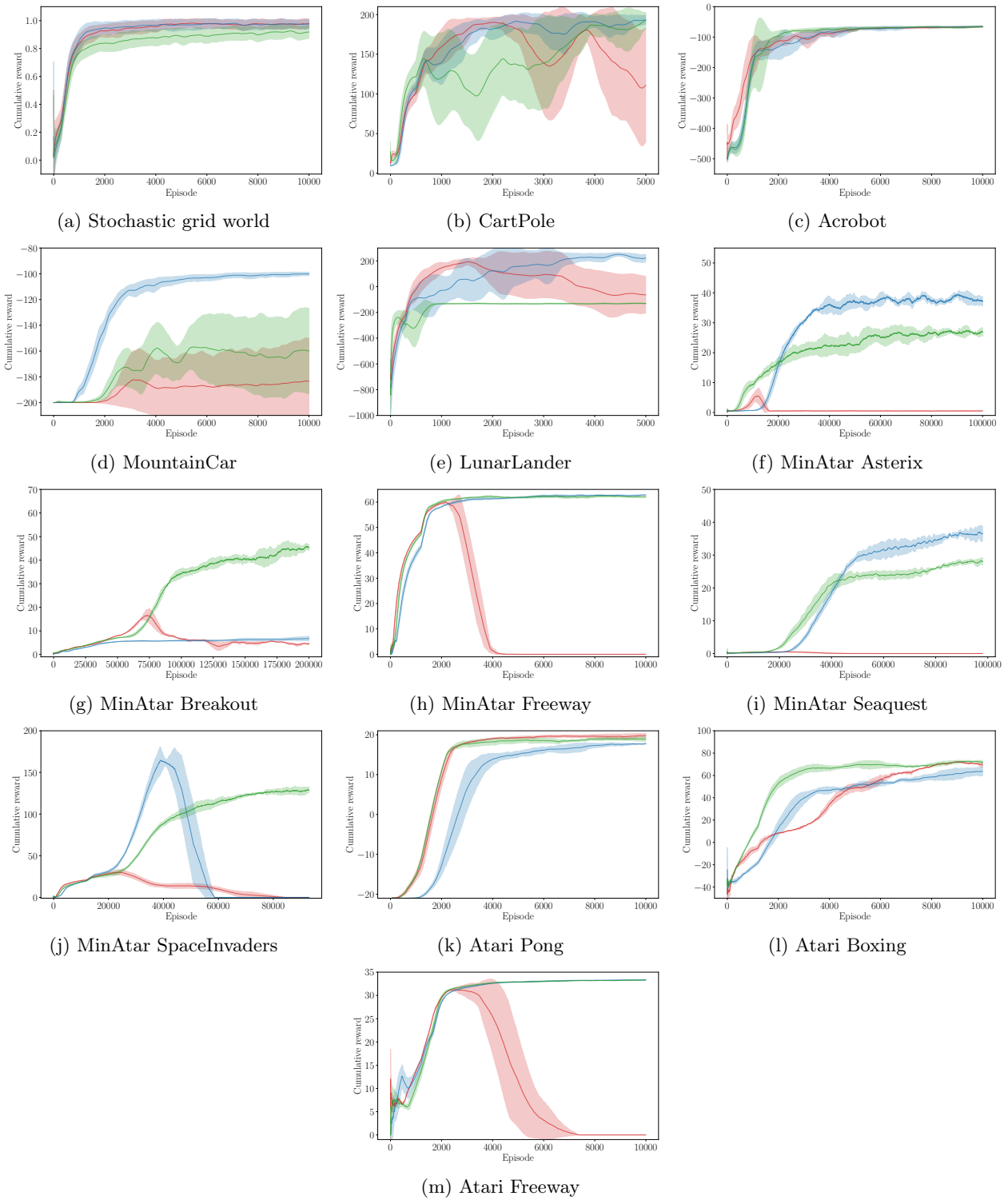


Figure 4.12: Performance of the UMDQN algorithm for the benchmark environments studied.

the loss defined in Equation (4.10), whose effect is not yet sufficiently understood for the different distribution representations and probability metrics, but also potentially depending on the control problem itself. This is an important open research question for distributional RL. Therefore, based on these observations, the distribution representation of the random return together with the probability metric should ideally be hyperparameters to be tuned depending on the environment and the control problem at hand. This claim contrasts with the current trend observed in distributional RL research, with the focus being mainly set on the QF and Wasserstein distance, as illustrated by the recent QR-DQN, IQN and FQF algorithms. For this reason, the present research work calls for a reconsideration of all distribution representations and probability metrics for future research in distributional RL.

UMDQN-KL algorithm. Even though the distributional Bellman operator $\mathcal{T}^\pi$ has been proven theoretically to not be a contraction mapping in the KL divergence, Figures 4.10 and 4.12 empirically show that this probability metric can still lead to the learning of both valuable decision-making policies and relevant random return probability distributions. This important observation suggests that the contraction property is not a necessary condition for converging towards the correct random return probability distribution in practice. Still, the learning process of the UMDQN-KL algorithm has been observed to be fairly less stable compared to other distributional RL algorithms. For several benchmark environments, the learning process may even suddenly stop with the performance entirely collapsing without subsequent recovery, as illustrated in Figure 4.12m. Additional experiments suggest that the occurrence of this problematic behaviour for a given environment is strongly tied to the domain $\mathcal{X}$ specified as hyperparameter (lower and upper bounds). A too restrained domain inevitably leads to truncated and hence incorrect probability distributions for the random return Z^π. On the contrary, if the domain is too wide, it may lead to numerical instabilities in the regions of the domain with almost no mass due to the definition of the KL divergence ($\lim_{x \to 0^+} \log(x) = -\infty$). Nevertheless, appropriately setting the hyperparameters associated with this domain $\mathcal{X}$ may be a particularly challenging task since it is strongly dependent on the control problem and because the probability distributions of the random return Z^π may significantly vary with different state-action pairs (s, a) as well as during the learning process. Another interesting observation about the UMDQN-KL algorithm is related to the asymmetry of the KL divergence ($\mathcal{L}_{KL}(A, B) \neq \mathcal{L}_{KL}(B, A)$). Empirically, the learning of valuable policies is observed with the loss $\mathcal{L}_{KL}(\mathcal{T}^\pi Z^\pi, Z^\pi)$ but not with $\mathcal{L}_{KL}(Z^\pi, \mathcal{T}^\pi Z^\pi)$.

UMDQN-C algorithm. Although this second distributional RL algorithm also requires the specification of hyperparameters associated with the domain $\mathcal{X}$, it is empirically observed to be far more stable and performing compared to the UMDQN-KL algorithm. This behaviour may potentially be explained by the distributional Bellman operator $\mathcal{T}^\pi$ being a contraction in the Cramer distance, but also by the fact that the loss to learn from is symmetric and does not numerically explode around regions of the domain with no probability density. Still, a relevant domain $\mathcal{X}$ has to be specified to expect reliable and satisfying results from the UMDQN-C algorithm, meaning that the full range of the returns has to be rigorously approximated beforehand. This requirement is probably the main weakness of this particular distributional RL algorithm. In Figure 4.12, relevant domains $\mathcal{X}$ are adopted to ensure a fair

and interesting comparison. The UMDQN-C algorithm may be the top-performing solution at first glance, but its performance inevitably decreases with less accurate domains.

UMDQN-W algorithm. Regarding the performance of the decision-making policies learnt, this third distributional RL algorithm may probably be the most versatile of the UMDQN algorithms, for two reasons. Firstly, the distributional Bellman operator $\mathcal{T}^{\pi}$ is a contraction mapping in the Wasserstein distance. Secondly, learning the QF of the random return Z^{π} does not require the challenging specification of the returns domain $\mathcal{X}$, since the QF takes inputs bounded in the range $[0, 1]$. Nevertheless, as previously explained in this section, the UMDQN-W algorithm is no longer an acceptable solution when it comes to the accuracy of the probability distributions learnt. Consequently, this distributional RL algorithm should only be considered for learning decision-making policies maximising the expectation of the random return, but not taking advantage of the full probability distributions.

Comparison with state-of-the-art distributional RL algorithms

To end the discussion of the results, the policy performance achieved by the UMDQN algorithm is briefly compared with that of the state-of-the-art distributional RL algorithms. More precisely, the DQN [55], CDQN [73], QR-DQN [81], IQN [82] and FQF [83] are evaluated alongside the three versions of the UMDQN algorithm on the benchmark environments previously described in Section 4.5.1. This additional analysis is solely provided for the sake of completeness, but does not represent a core contribution of the present research work. Indeed, the primary objective of this research is to empirically derive new insights about the distributional RL approach, but not to introduce a novel distributional RL algorithm outperforming the state-of-the-art algorithmic solutions on a given testbench.

This empirical comparison on policy performance is summarised in Figure 4.13 hereafter. In a similar way to the former plots, the results are averaged over five different random seeds for better reliability, and post-processed using the moving average technique to further smooth the curves. Nevertheless, for the sake of readability, the performance variance of the distributional RL algorithms is no longer depicted on the plots.

Since the full Atari-57 benchmark [53] has not been taken into consideration for evaluating the novel UMDQN distributional RL algorithm proposed, this research work does not make any claim regarding the top-performing solution for this particular benchmark. However, it is enlightening to observe that the UMDQN algorithm achieves an impressive performance which is on par with the state-of-the-art distributional RL algorithms on the benchmark environments adopted in this research work. Indeed, the UMDQN algorithm consistently ranks in the top three in terms of decision-making policy performance for all these control problems. In the author's opinion, this result consolidates the soundness of the proposed approach together with the relevance of the conclusions drawn by this research work.

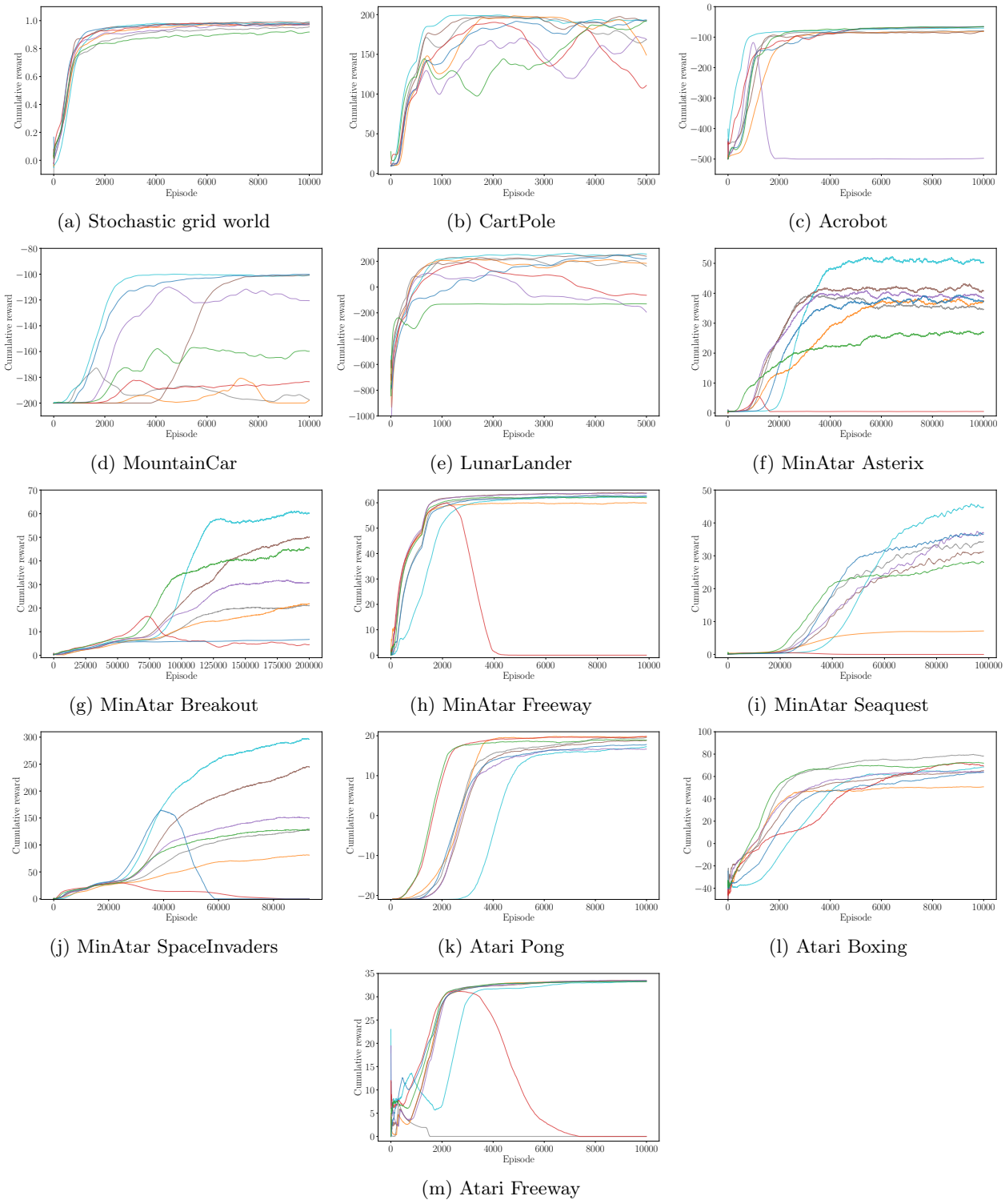


Figure 4.13: Comparison of the UMDQN algorithm with the state-of-the-art distributional RL algorithms.

4.6 Conclusions

This fourth thesis chapter introduces the *unconstrained monotonic deep Q-network* (UMDQN) distributional RL algorithm, by combining a new methodology for learning the probability distribution of the random return independently of its representation with the UMNN architecture for modelling these distributions. Taking advantage of some properties of this novel distributional RL algorithm, the experiments conducted yield three important observations. Firstly, the choice of the probability distribution representation coupled with the choice of the probability metric have to ideally be dependent on the control problem, since no clear winner could be identified among the three versions of the UMDQN algorithm for the set of benchmark environments studied. This result contrasts with the current dominant trend in distributional RL research, that primarily focuses on the QF and Wasserstein distance. Secondly, the methodology adopted by several state-of-the-art algorithms for learning the QF of the random return involves an important approximation, which results in the learning of inaccurate probability distributions. This approach remains totally sound when attempting to learn decision-making policies maximising the expectation of the random return. On the contrary, it should be discarded when aiming to take advantage of other characteristics of the random return distribution, for instance with risk-sensitive policies. Thirdly, the contraction mapping property for the distributional Bellman operator is not a necessary condition to learn the correct probability distribution of the random return, but may still be beneficial. This observation highlights the existing gap between theory and practice in distributional RL, and encourages future research on the distributional Bellman operator as well as on the convergence of distributional RL algorithms in general.

To conclude this thesis chapter, several avenues for future work are suggested. Firstly, the gap highlighted in this research work between theory and practice in distributional RL could be narrowed by deriving theoretical guarantees and properties for the UMDQN algorithm. Secondly, building on the qualitative analysis of the probability distributions learnt that is conducted in this research work, an innovative performance assessment methodology could be designed to quantitatively evaluate the accuracy of the random return distributions learnt by a distributional RL algorithm, independently of the resulting policy performance. Thirdly, the approximation in Equation 4.16 for the learning of the QF based on the distributional Bellman equation deserves more research, in order to acquire a better understanding of the problem and potentially find an alternative solution. Fourthly, the performance achieved by the UMDQN algorithm is expected to be significantly improved by implementing the enhancements from the Rainbow algorithm [56]: multi-step learning [57], double Q-learning [58], prioritised experience replay [59], duelling architecture [60] and noisy networks [62]. Lastly, a promising evolution of the UMDQN algorithm could be to concurrently manage different distribution representations and probability metrics, and intelligently combine this information to further improve the performance of the distributional RL algorithm.

Nothing in life is to be feared, it is only to be understood. Now is the time to understand more, so that we may fear less.

— Marie Curie

Chapter 5

Risk-Sensitive Policy with Distributional Reinforcement Learning

Figure 5.1: Illustration of Chapter 5 that is entitled *Risk-Sensitive Policy with Distributional Reinforcement Learning*, created by a generative art AI [1].

Chapter overview

Classical reinforcement learning techniques are generally concerned with the design of decision-making policies driven by the maximisation of the expected outcome. Nevertheless, this approach does not take into consideration the eventual risk that is associated with the actions taken, which may be critical in certain applications. To address that issue, the present research work introduces a novel methodology based on distributional reinforcement learning to derive sequential decision-making policies that are sensitive to the risk, the latter being modelled by the tail of the return probability distribution. The core idea is to replace the Q function generally standing at the core of learning schemes in RL by another function taking into account both the expected return and the risk. Named *risk-based utility function* U, it can be extracted from the random return distribution Z that is naturally learnt by any distributional RL algorithm. This enables the spanning of the complete potential trade-off between risk minimisation and expected return maximisation, in contrast to fully risk-averse methodologies. Fundamentally, this thesis chapter presents a truly practical and accessible solution for learning risk-sensitive policies with minimal modification to the original distributional RL algorithm, and with an emphasis on the interpretability of the resulting decision-making process.

This thesis chapter is primarily based on the following scientific publication [5]:

Thibaut Théate and Damien Ernst. Risk-Sensitive Policy with Distributional Reinforcement Learning. *Algorithms*, 16(7):325, 2023.

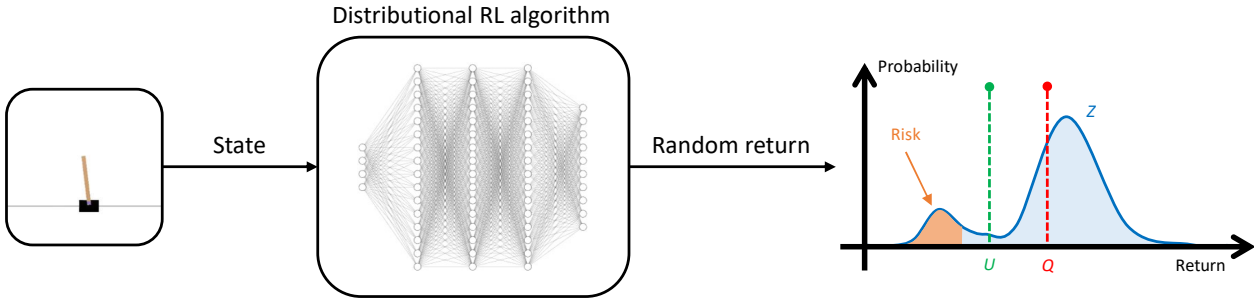


Figure 5.2: General illustration of the novel technical solution presented in this thesis chapter entitled *Risk-Sensitive Policy with Distributional Reinforcement Learning*.

5.1 Introduction

As previously explained in this thesis, the reinforcement learning (RL) approach is concerned with the learning process of a sequential decision-making policy on the basis of interactions between an agent and its environment [64]. More precisely, the training is based on the rewards acquired by the agent from the environment as a consequence of its actions. Within that particular context, the objective is to identify the actions maximising the discounted sum of rewards, which is also named return. There exist multiple sound approaches based on the RL paradigm, and key successes/milestones have been achieved throughout the years of research. Nevertheless, these RL algorithms present the key weakness of mostly relying on the expectation of the return, not on its complete probability distribution. For instance, the popular *Q-learning* approach is based on the modelling of the Q function, which can in fact be seen as an estimation of the expected return [51].

While focusing exclusively on the expectation of the return has proven to be perfectly sound for numerous applications, this approach however exhibits clear limitations for other decision-making problems. Indeed, some areas of application may also require to properly mitigate the risk associated with the actions taken [93]. Market environments, on which the present doctoral thesis focuses, belong to this category. Additionally, one could for instance mention the healthcare [94] and finance [2] sectors, but also robotics in general [95] especially autonomous driving [96]. Such a requirement for risk management may not only be true for the decision-making policy, but potentially also for the exploration policy during the learning process. Moreover, properly taking into consideration the risk may be particularly convenient in environments characterised by substantial uncertainty.

The present research work suggests taking advantage of the distributional RL approach, belonging to the *Q-learning* category, in order to learn risk-sensitive policies. As previously explained in Chapter 4, a distributional RL algorithm focuses on the complete probability distribution of the random return rather than only its expectation [61]. Such an approach presents three key advantages. Firstly, it enables the learning of a richer representation of the environment, which may lead to an increase in the decision-making policy performance. Secondly, the distributional RL approach contributes to improve the explainability of the decision-making process, which is critical in machine learning to avoid black-box models. Lastly and most importantly for this research work, it makes the convenient derivation of decision-making policies but also exploration strategies that are sensitive to the risk possible.

The core idea promoted by this research work is the use of the *risk-based utility function*, denoted U, as a replacement for the popular Q function for action selection. In fact, it may be seen as an extension of the Q function taking into consideration the risk, which is assumed to be represented by the worst returns achievable by a policy. Therefore, the function U is to be derived from the complete probability distribution of the random return Z, which is learnt by any distributional RL algorithm. The single modification to that RL algorithm to learn risk-sensitive policies is to employ the utility function U rather than the expected return Q for both exploration and decision-making. This enables the proposed approach to become a very practical and interpretable RL solution for achieving risk-sensitive decision making.

5.2 Literature review

The core objective of the classical RL approach is to learn optimal decision-making policies without any concerns about risk or safety [64]. In this generic case, the resulting policies are said to be *risk-neutral*. Nevertheless, as previously explained, there are numerous real-world applications requiring to take into consideration the risk in order to ensure safer decision making [93]. In fact, two main methodologies can be identified for achieving safe RL. Firstly, the optimality criterion pursued can be modified so that a safety factor is included. Secondly, the exploration process during the training phase can be altered based on a risk metric [97]. These techniques give rise to *risk-sensitive* or *risk-averse* policies.

Scientific research on risk-sensitive RL has been particularly active for the past decade. Various relevant risk criteria have been studied for that purpose. The most popular ones are undoubtedly the *mean-variance* [98, 99, 100] and the *(Conditional) Value at Risk* (CVaR) [101, 102, 103]. Innovative techniques have been introduced for both *policy gradient* [104, 105, 106] and *value iteration* [107, 108, 109, 110, 111] approaches, with the solutions proposed covering both discrete and continuous action spaces. Additionally, risk-sensitive methodologies have also been investigated in certain niche sub-fields of RL, such as robust adversarial RL [112], but also multi-agent RL [113].

Focusing on the *value iteration* methodology, the novel distributional RL approach [61] has been a key breakthrough, by giving access to the full probability distribution of the random return. To begin with, [108] suggests to achieve risk-sensitive decision-making via a distortion risk measure. Applied on top of the IQN distributional RL algorithm, this is in fact equivalent to changing the sampling distribution of the quantiles. In such a setting, assigning more weight to lower quantiles results in a more risk-averse objective. In [109], a novel actor-critic framework is presented, based on the distributional RL approach for the critic component. The latter work is extended to the offline setting in [110], since training RL agents online may be prohibitive because of the risk inevitably induced by exploration. One can finally mention [111] that introduces the *Worst-Case Soft Actor Critic* (WCSAC) algorithm, which is based on the approximation of the probability distribution of accumulated safety-costs in order to achieve risk control. More precisely, a certain level of CVaR, estimated from the distribution, is regarded as a safety constraint.

In light of this scientific literature, the novel solution introduced in this research work presents key advantages. Firstly, the methodology proposed is relatively simple and can be applied on top of any distributional RL algorithm with minimal modification to the original algorithm. Secondly, the proposed approach enables to span the entire potential trade-off between risk minimisation and expected return maximisation. Indeed, according to the user's needs, the decision-making policy learnt can be risk-averse, risk-neutral or in between the two (risk-sensitive). Lastly, the new solution presented contributes to improve the interpretability of the decision-making policy learnt.

5.3 Theoretical background

5.3.1 Markov decision process

Traditionally in RL, the interactions between the agent and its environment are modelled as a *Markov decision process* (MDP). In fact, an MDP is a 6-tuple $(\mathcal{S},\ \mathcal{A},\ p_R,\ p_T,\ p_0,\ \gamma)$ where $\mathcal{S}$ and $\mathcal{A}$ respectively are the state and action spaces, $p_R(r|s,a)$ is the probability distribution from which the reward $r \in \mathbb{R}$ is drawn given a state-action pair (s,a), $p_T(s'|s,a)$ is the transition probability distribution, $p_0(s_0)$ is the probability distribution over the initial states $s_0 \in \mathcal{S}$, and $\gamma \in [0,1[$ is the discount factor. The RL agent makes some decisions according to its policy $\pi : \mathcal{S} \to \mathcal{A}$, which is assumed deterministic in this research work, mapping the states $s \in \mathcal{S}$ to the actions $a \in \mathcal{A}$.

5.3.2 Distributional reinforcement learning

As previously explained in Chapter 4, in classical *Q-learning* RL, the core idea is to model the *state-action value function* $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ of a policy π. This important quantity $Q^\pi(s,a)$ represents the expected discounted sum of rewards obtained by executing an action $a \in \mathcal{A}$ in a state $s \in \mathcal{S}$ and then following a policy π:

$$Q^\pi(s,a) = \mathop{\mathbb{E}}_{s_t,r_t}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right], \qquad (s_0,a_0) := (s,a),\ a_t = \pi(s_t). \tag{5.1}$$

The key to the learning process is the *Bellman equation* [52], that the Q function satisfies:

$$Q^\pi(s,a) = \mathop{\mathbb{E}}_{s',r}\left[r + \gamma Q^\pi(s', \pi(s'))\right]. \tag{5.2}$$

In classical RL, the main objective is to determine an *optimal policy* π^* which can be defined based on the *optimal state-action value function* $Q^* : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ as follows:

$$Q^*(s,a) = \mathop{\mathbb{E}}_{s',r}\left[r + \gamma \max_{a' \in \mathcal{A}} Q^*(s',a')\right], \tag{5.3}$$

$$\pi^*(s) \in \operatorname*{argmax}_{a \in \mathcal{A}}\ Q^*(s,a)\ . \tag{5.4}$$

The optimal policy π^* maximises the expected return (discounted sum of rewards). In this research work, an alternative objective criterion will be presented in the next section for achieving optimality in a risk-sensitive RL setting.

The distributional RL approach goes a step further by modelling the complete probability distribution over returns instead of only its expectation [61], as illustrated in Figure 5.3. To this end, let the reward $R(s,a)$ be a random variable distributed under $p_R(\cdot|s,a)$, the *state-action value distribution* $Z^\pi \in \mathcal{Z}$ (also called *state-action return distribution function*) of a policy π is a random variable defined as the following:

$$Z^\pi(s,a) \overset{D}{=} \sum_{t=0}^{\infty} \gamma^t R(s_t,a_t)\ , \qquad (s_0,a_0) := (s,a),\ a_t = \pi(s_t),\ s_{t+1} \sim p_T(\cdot|s_t,a_t), \tag{5.5}$$

where $A \overset{D}{=} B$ denotes the equality in probability distribution between the random variables A and B. Consequently, the state-action value function Q^π is the expectation of the *random return* Z^π. Equivalent to the expected case, there exists a *distributional Bellman equation* that recursively describes the random return Z^π of interest:

$$Z^\pi(s,a) \overset{D}{=} R(s,a) + \gamma P^\pi Z^\pi(s,a) \ , \tag{5.6}$$

$$P^\pi Z^\pi(s,a) :\overset{D}{=} Z^\pi(s',a') \ , \qquad s' \sim p_T(\cdot|s,a), \ a' = \pi(s'), \tag{5.7}$$

where $P^\pi : \mathcal{Z} \to \mathcal{Z}$ is the transition operator. Finally, one can define the *distributional Bellman operator* $\mathcal{T}^\pi : \mathcal{Z} \to \mathcal{Z}$ together with the *distributional Bellman optimality operator* $\mathcal{T}^* : \mathcal{Z} \to \mathcal{Z}$ as follows:

$$\mathcal{T}^\pi Z^\pi(s,a) \overset{D}{=} R(s,a) + \gamma P^\pi Z^\pi(s,a) \ , \tag{5.8}$$

$$\mathcal{T}^* Z^*(s,a) \overset{D}{=} R(s,a) + \gamma Z^*\left(s', \pi^*(s')\right), \qquad s' \sim p_T(\cdot|s,a). \tag{5.9}$$

As thoroughly explained in Chapter 4, a distributional RL algorithm may be characterised by two core features. Firstly, both representation and parameterisation of the random return probability distribution have to be selected. There exist multiple solutions for representing a unidimensional distribution, such as the probability density function (PDF), cumulative distribution function (CDF) or quantile function (QF). In practice, deep neural networks (DNNs) are generally used for the approximation of these particular functions. The second key feature relates to the probability metric adopted for comparing two distributions, such as the Kullback-Leibler (KL) divergence, the Cramer distance or the Wasserstein distance. More precisely, the role of the probability metric in distributional RL is to quantitatively compare two probability distributions of the random return so that a temporal difference (TD) learning method is applied, in a similar way to the mean squared error between Q values in classical RL. Moreover, the probability metric plays an even more important role as different metrics offer distinct theoretical convergence guarantees for distributional RL.

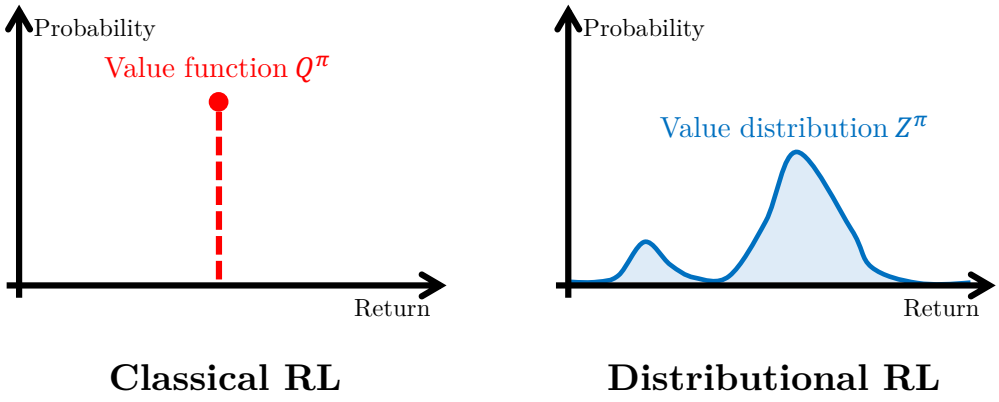


Figure 5.3: Intuitive graphical comparison between classical RL and distributional RL.

5.4 Methodology

5.4.1 Objective criterion for risk-sensitive RL

As explained in the previous section, the objective in classical RL is to learn a decision-making policy $\pi \in \Pi$ that maximises in expectation the discounted sum of rewards. Formally, this objective criterion can be expressed as the following:

$$\underset{\pi}{\text{maximise}}\ \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]. \tag{5.10}$$

In order to effectively take into consideration the risk and value its mitigation, this research work presents an update for the former objective. In fact, coming up with a generic definition for the risk is not trivial since the risk is generally dependent on the decision-making problem. In the present research work, it is assumed that the risk is assessed on the basis of the worst returns achievable by a policy π. Therefore, a successful decision-making policy should ideally maximise the expected discounted sum of rewards while avoiding low values for the worst-case returns. The latter requirement is approximated with a new constraint attached to the former objective defined in Equation (5.10): the probability of having the policy achieving a return lower than a certain minimum value should not exceed a given threshold. Mathematically, the alternative objective criterion proposed for risk-sensitive RL can be expressed as follows:

$$\begin{aligned} &\underset{\pi}{\text{maximise}} \quad \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right], \\ &\text{such that} \quad p\left[\sum_{t=0}^{\infty} \gamma^t r_t \leq R_{\min}\right] \leq \epsilon\ , \end{aligned} \tag{5.11}$$

where:

- $p[\star]$ denotes the probability of the event $\star$,
- $R_{\min}$ is the minimum acceptable return (from a risk mitigation perspective),
- $\epsilon \in [0, 1]$ is the threshold probability not to be exceeded.

5.4.2 Practical modelling of the risk

As previously hinted, this research work assumes that the risk associated with a certain action is related to the worst achievable returns when executing that particular action and then following a given decision-making policy π. In such a context, the distributional RL approach becomes particularly interesting, by providing access to the full probability distribution of the random return Z^π. Therefore, the risk can be efficiently assessed by examining the so-called tail of the learnt probability distribution. Moreover, the constraint in Equation (5.11) can be approximated through popular recognised risk measures such as the *Value at Risk* (VaR) and *Conditional Value at Risk* (CVaR). Illustrated in Figure 5.4, these two risk measures are mathematically expressed as the following:

$$\mathrm{VaR}_\rho(Z^\pi) = \inf\{z \in \mathbb{R} : F_{Z^\pi}(z) \geq \rho\}, \tag{5.12}$$

$$\mathrm{CVaR}_\rho(Z^\pi) = \mathbb{E}\left[z \mid z \leq \mathrm{VaR}_\rho(Z^\pi)\right], \tag{5.13}$$

where F_{Z^π} represents the CDF of the random return Z^π.

More generally, this research work introduces the *state-action risk function* $R^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ of a decision-making policy π, which is the equivalent of the Q function for the risk. More precisely, that function $R^\pi(s,a)$ quantifies the riskiness of the discounted sum of rewards obtained by executing an action $a \in \mathcal{A}$ in a state $s \in \mathcal{S}$ and then following a policy π:

$$R^\pi(s,a) = \mathcal{R}_\rho\left[Z^\pi(s,a)\right], \tag{5.14}$$

where:

- $\mathcal{R}_\rho : \mathcal{Z} \to \mathbb{R}$ is a function extracting risk features from the random return probability distribution Z^π, such as VaR_ρ or CVaR_ρ,
- $\rho \in]0,1[$ is a parameter corresponding to the cumulative probability associated with the worst returns, generally between 0% and 10%. In other words, this parameter controls the size of the random return distribution tail from which the risk is estimated.

5.4.3 Risk-based utility function

In order to pursue the objective criterion defined in Equation (5.11) for risk-sensitive RL, this research work introduces a new concept: the *state-action risk-based utility function*. Denoted $U^\pi(s,a)$, the utility function assesses the quality of an action $a \in \mathcal{A}$ in a certain state $s \in \mathcal{S}$, in terms of expected performance and risk, assuming that the policy π is followed afterwards. In fact, the intent is to extend the popular Q function so that the risk is taken into consideration, by taking advantage of the risk function defined in Section 5.4.2. More precisely, the utility function U^π is built as a linear combination of the Q^π and R^π functions.

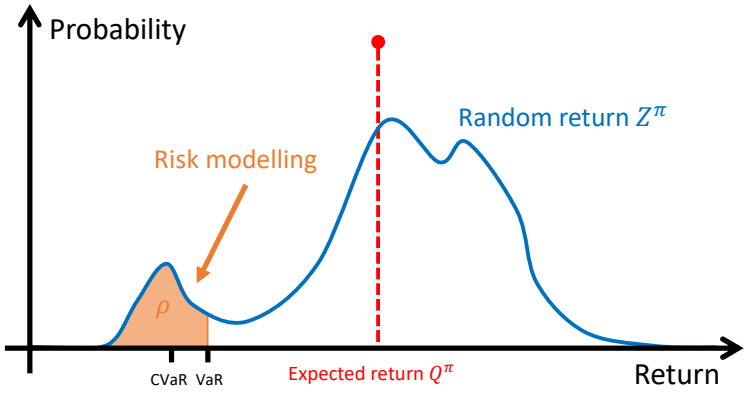


Figure 5.4: Illustration of the risk modelling adopted in this research work, on the basis of the probability distribution of the random return Z^π learnt by a distributional RL algorithm.

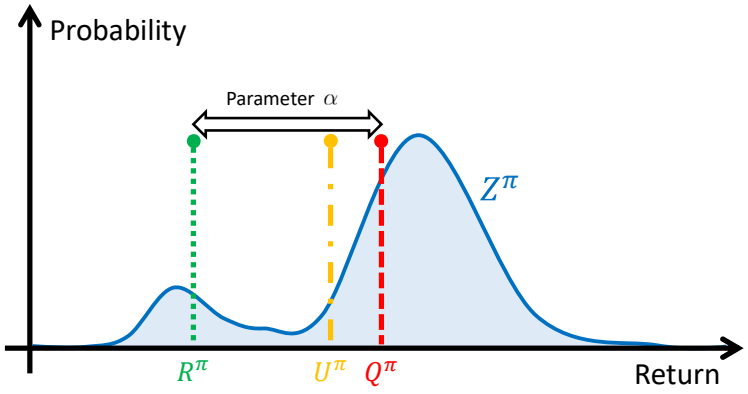


Figure 5.5: Illustration of the risk-based utility function U^π for a random return probability distribution, with the parameter $\alpha = 0.75$ in this case.

Formally, the risk-based utility function $U^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ of a decision-making policy π is defined as the following:

$$U^\pi(s,a) = \alpha\ Q^\pi(s,a)\ +\ (1-\alpha)\ R^\pi(s,a) \tag{5.15}$$

$$= \alpha\ \mathbb{E}\left[Z^\pi(s,a)\right]\ +\ (1-\alpha)\ \mathcal{R}_\rho\left[Z^\pi(s,a)\right], \tag{5.16}$$

where $\alpha \in [0,1]$ is a parameter controlling the trade-off between expected performance and risk. If $\alpha = 0$, the utility function will be maximised with a fully risk-averse decision-making policy. On the contrary, if $\alpha = 1$, the utility function degenerates into the Q function quantifying the performance on expectation (risk-neutral). Figure 5.5 graphically describes the newly introduced risk-based utility function U^π, which moves along the x-axis between the quantities R^π and Q^π when modifying the value of the parameter α.

5.4.4 Risk-sensitive distributional RL algorithm

In most applications, the motivation for choosing the distributional RL approach over the classical one is related to the improved expected performance that results from the learning of a richer representation of the environment. Despite having access to the full probability distribution of the random return, only the expectation is exploited to derive policies:

$$\pi(s) \in \underset{a\in\mathcal{A}}{\operatorname{argmax}}\ \underbrace{\mathbb{E}\left[Z^\pi(s,a)\right]}_{Q^\pi(s,a)}. \tag{5.17}$$

Nevertheless, the random return Z^π does also contain valuable information about the risk, which could be exploited to learn risk-sensitive decision-making and exploration policies. The present research work suggests to achieve risk-sensitive distributional RL by maximising the utility function U^π, derived from Z^π, instead of the expected return Q^π when selecting actions. This alternative operation would be performed during both exploration and exploitation. Even though maximising the utility function is not exactly equivalent to the optimisation of the objective criterion defined in Equation (5.11), it is a relevant step towards achieving risk-sensitive RL. Consequently, a risk-sensitive policy π can be derived as follows:

$$\pi(s) \in \underset{a\in\mathcal{A}}{\operatorname{argmax}}\ U^\pi(s,a)\ . \tag{5.18}$$

Algorithm 7 Risk-sensitive distributional RL algorithm

Initialise the experience replay memory M of capacity C.
Initialise both the main and target DNN weights $\theta = \theta^-$.
for episode $= 0$ **to** N **do**
  for time step $t = 0$ **to** T, or until episode termination **do**
    Acquire the state s from the environment $\mathcal{E}$.
    With probability ϵ, select a random action $a \in \mathcal{A}$.
    Otherwise, select the action $a = \text{argmax}_{a' \in \mathcal{A}} U^\pi(s, a'; \theta)$.
    Execute action a in environment $\mathcal{E}$ to get the next state s' and the reward r.
    Store the experience $e = (s, a, r, s')$ in M.
    Randomly sample from M a minibatch of N_e experiences $e_i = (s_i, a_i, r_i, s_i^{'})$.
    for $i = 0$ **to** N_e **do**
      Distributional Bellman equation: $Z^\pi(s_i, a_i; \theta) \overset{D}{=} r_i + \gamma Z^\pi(s_{i+1}, \text{argmax}_{a'_i \in \mathcal{A}} U^\pi(s_{i+1}, a'_i; \theta^-); \theta^-)$.
    end for
    Compute the resulting loss $\mathcal{L}(\theta)$, according to the probability metric selected.
    Update the main DNN parameters θ using a deep learning optimiser with learning rate L_r.
    Update the target DNN parameters $\theta^- = \theta$ every N^- steps.
    Anneal the ϵ-greedy exploration parameter ϵ.
  end for
end for

Throughout the learning phase, a classical Q-learning algorithm is expected to progressively converge towards the optimal value function Q^* that naturally arises from the optimal decision-making policy π^*. In a similar way, the proposed risk-sensitive RL algorithm jointly learns the optimal policy π^* and the *optimal state-action risk-based utility function* U^*. More formally, the latter two are mathematically defined as the following:

$$U^*(s, a) = \alpha \, \mathbb{E}\left[Z^*(s, a)\right] \; + \; (1 - \alpha) \, \mathcal{R}_\rho\left[Z^*(s, a)\right], \tag{5.19}$$

$$Z^*(s, a) \overset{D}{=} R(s, a) + \gamma Z^*\left(s', \pi^*(s')\right), \tag{5.20}$$

$$\pi^*(s) \in \underset{a \in \mathcal{A}}{\text{argmax}} \; U^*(s, a) \; . \tag{5.21}$$

The novel methodology proposed by this research work to effectively learn risk-sensitive decision-making policies based on the distributional RL approach is summarised as follows. Firstly, select any distributional RL algorithm that learns the full probability distribution of the random return Z^π. Secondly, leave the learning process unchanged except for action selection, which involves the maximisation of the utility function U^π rather than the expected return Q^π. This adaptation is the single change to the distributional RL algorithm, occurring at two different locations within the algorithm: *i.* the generation of new experiences by interacting with the environment, *ii.* the learning of the random return Z^π based on the distributional Bellman equation. However, this adaptation has no consequence on the random return Z^π itself learnt by the distributional RL algorithm, only on the actions derived from that probability distribution. To end this section, Algorithm 7 details the proposed solution in a generic way, with the required modifications being highlighted in colour.

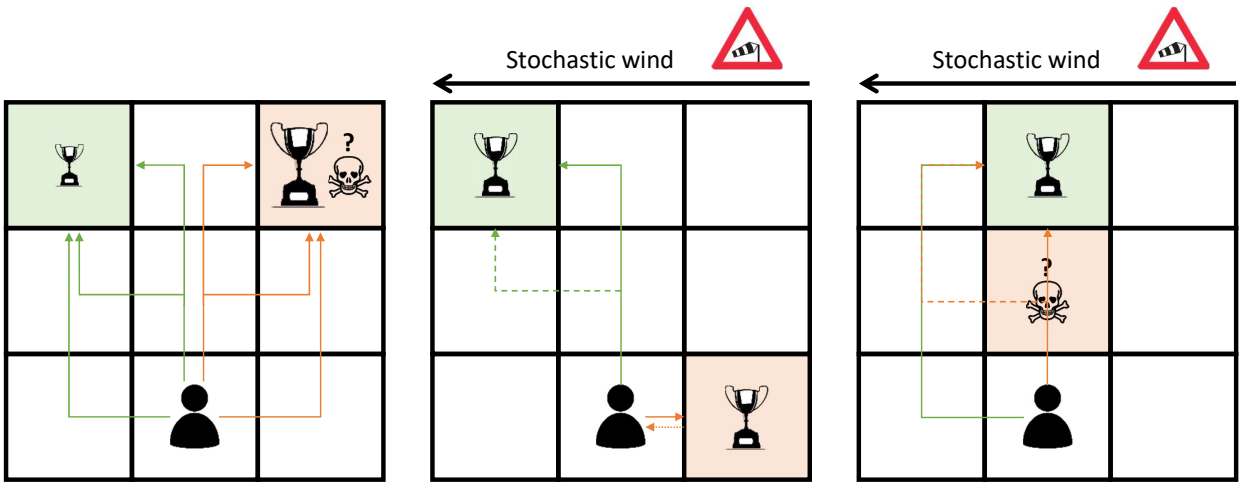
5.5 Performance assessment methodology

5.5.1 Benchmark environments

This research work introduces some novel benchmark environments to assess the soundness of the proposed methodology to design risk-sensitive policies based on the distributional RL approach. These environments consist of three toy problems that are specifically designed to highlight the importance of taking into consideration the risk for a decision-making policy. More precisely, the control problems are built in such a way that the optimal policy will differ depending on whether the objective is to solely maximise the expected performance or to also mitigate the risk. This behaviour is achieved by including relevant stochasticity in both the state transition function p_T and the reward function p_R. Moreover, the benchmark environments are designed to be relatively straightforward in order to ease the analysis and understanding of the decision-making policies learnt. This simplicity ensures the accessibility of the experiments as well, since distributional RL algorithms generally require a considerable amount of computing power. Figure 5.6 illustrates these three benchmark environments, and highlights the optimal paths to be learnt depending on the objective pursued.

The first benchmark environment presented is named *risky rewards*. It consists of a 3×3 grid world within which an agent has to reach one of two objective areas, that are equidistant from its fixed initial location. The difficulty of this control problem lies in the choice of the objective area to target, because of the stochasticity present in the reward function. Reaching the first objective area yields a reward with a lower value in expectation and a limited deviation from that average. On the contrary, reaching the second objective location yields a reward that is higher in expectation, at the cost of an increased risk. Formally, the underlying MDP can be described as the following:

- $\mathcal{S} \in \{0, 1, 2\} \times \{0, 1, 2\}$, a state s being composed of the two coordinates of the agent within the grid,
- $\mathcal{A} = \{\texttt{RIGHT}, \texttt{DOWN}, \texttt{LEFT}, \texttt{UP}\}$, with an action a being a moving direction,
- $p_R(r|s, a) \sim \mathcal{N}(\mu, \sigma^2)$ where:
  - $\mu = 0.3$ and $\sigma = 0.1$ if the agent reaches the first objective location (terminal state),
  - $\mu = 1.0$ and $\sigma = 0.1$ with a 75% chance, and $\mu = -1.0$ and $\sigma = 0.1$ with a 25% chance if the agent reaches the second objective location (terminal state),
  - $\mu = -0.1$ and $\sigma = 0.1$ otherwise,
- $p_T(s'|s, a)$ associates a 100% chance to move once in the chosen direction, while keeping the agent within the 3×3 grid world (crossing a border is prohibited),
- p_0 associates a probability of 1 to the state $s = [1, 0]$, which is the position of the agent depicted in Figure 5.6,
- $\gamma = 0.9$.



(a) Risky rewards (b) Risky transitions (c) Risky grid world

Figure 5.6: Illustration of the benchmark environments introduced in this research work for the performance assessment of risk-sensitive decision-making policies. The optimal objective locations/paths in terms of risk mitigation and expected return maximisation are respectively highlighted in green and orange.

The second benchmark environment studied is named *risky transitions*. It consists of a 3×3 grid world within which an agent has to reach one of two objective areas as quickly as possible, in the presence of a stochastic wind. The agent is initially located in a fixed area that is very close to an objective, but the required move to reach that goal is in opposition to the wind direction. Following that path results in a reward that is higher in expectation, but there is a risk to be repeatedly countered by the stochastic wind. On the contrary, the longer path is safer but yields a lower reward on average. For the sake of completeness, the underlying MDP can be described as the following:

- $\mathcal{S} \in \{0, 1, 2\} \times \{0, 1, 2\}$, a state s being composed of the two coordinates of the agent within the grid,
- $\mathcal{A} = \{\texttt{RIGHT}, \texttt{DOWN}, \texttt{LEFT}, \texttt{UP}\}$, with an action a being a moving direction,
- $p_R(r|s, a) \sim \mathcal{N}(\mu, \sigma^2)$ where:
  - $\mu = 1.0$ and $\sigma = 0.1$ if the agent reaches an objective location (terminal state),
  - $\mu = -0.3$ and $\sigma = 0.1$ otherwise,
- $p_T(s'|s, a)$ associates a 100% chance to move once in the chosen direction AND a 50% chance to get pushed once to the left by the stochastic wind, while keeping the agent within the 3×3 grid world (crossing a border is prohibited),
- p_0 associates a probability of 1 to the state $s = [1, 0]$, which is the position of the agent depicted in Figure 5.6,
- $\gamma = 0.9$.

The last benchmark environment presented is denominated *risky grid world*. This control problem can be viewed as a combination of the two environments previously described since it integrates both stochastic rewards and transitions. It consists once again of a 3×3 grid world within which an agent, initially located in a fixed area, has to reach a fixed objective location as quickly as possible. To achieve that goal, three paths are available. The agent may choose the shortest path to the objective location that is characterised by a stochastic trap, or get around this risky situation by taking a significantly longer route. This bypass can be done from the left or from the right, another critical choice in terms of risk because of the stochastic wind that may push the agent to the left. Once again, the optimal path is therefore dependent on the objective criterion to pursue. More precisely, the underlying MDP can be described as the following:

- $\mathcal{S} \in \{0, 1, 2\} \times \{0, 1, 2\}$, a state s being composed of the two coordinates of the agent,
- $\mathcal{A} = \{\texttt{RIGHT}, \texttt{DOWN}, \texttt{LEFT}, \texttt{UP}\}$, with an action a being a moving direction,
- $p_R(r|s, a) \sim \mathcal{N}(\mu, \sigma^2)$ where:
  - $\mu = 1.0$ and $\sigma = 0.1$ if the agent reaches the objective location (terminal state),
  - $\mu = -0.2$ and $\sigma = 0.1$ with a 75% chance, and $\mu = -2.0$ and $\sigma = 0.1$ with a 25% chance if the agent reaches the stochastic trap location (terminal state),
  - $\mu = -0.2$ and $\sigma = 0.1$ otherwise,
- $p_T(s'|s, a)$ associates a 100% chance to move once in the chosen direction AND a 25% chance to get pushed once to the left by the stochastic wind, while keeping the agent within the 3×3 grid world (crossing a border is prohibited),
- p_0 associates a probability of 1 to the state $s = [1, 0]$, which is the position of the agent depicted in Figure 5.6,
- $\gamma = 0.9$.

5.5.2 Risk-sensitive distributional RL algorithm analysed

The distributional RL algorithm selected to assess the soundness of the proposed approach for learning risk-sensitive policies is the *Unconstrained Monotonic Deep Q-Network with Cramer* (UMDQN-C) [3] previously introduced in Chapter 4. Basically, this particular distributional RL algorithm models the CDF of the random return in a continuous way by taking advantage of the Cramer distance for deriving the TD-error. Additionally, the probability distributions learnt are ensured to be perfectly valid thanks to the specific architecture exploited to model the random return: *Unconstrained Monotonic Neural Network* (UMNN) [75]. The latter has been demonstrated to be a universal approximator of continuous monotonic functions, which is particularly convenient for representing CDFs. In Chapter 4, the UMDQN-C algorithm has been shown to achieve great results, both in terms of policy performance and in terms of random return probability distribution quality. This second feature motivates the selection of this specific distributional RL algorithm to conduct the following experiments, since accurate random return probability distributions are required to properly estimate the risk.

As previously explained in Section 5.4.2, the proposed methodology requires the choice of a relevant function $\mathcal{R}_\rho$ for extracting risk features from the random return probability distribution learnt Z^π. In the following experiments, the *Value at Risk* (VaR) is adopted to estimate the risk. This choice is motivated by both the popularity of that risk measure in practice and by the efficiency of computation. Indeed, this quantity can be conveniently derived from the CDF of the random return learnt by the UMDQN-C algorithm, as indicated mathematically in Equation (5.12).

The resulting novel risk-sensitive distributional RL algorithm is denominated *Risk-Sensitive Unconstrained Monotonic Deep Q-Network with Cramer* (RS-UMDQN-C). For the sake of clarity, Algorithm 8 provides its detailed pseudo-code. Basically, this new RL algorithm is the result of the application of the methodology described in Algorithm 7 to the UMDQN-C algorithm introduced in Chapter 4, with the Value at Risk being selected as risk measure.

Algorithm 8 RS-UMDQN-C algorithm

Initialise the experience replay memory M of capacity C.
Initialise the main UMNN weights θ (Xavier initialisation).
Initialise the target UMNN weights $\theta^- = \theta$.
for episode $= 0$ **to** N **do**
  for $t = 0$ **to** T, or until episode termination **do**
    Acquire the state s from the environment $\mathcal{E}$.
    With probability ϵ, select a random action $a \in \mathcal{A}$.
    Otherwise, select $a = \text{argmax}_{a' \in \mathcal{A}} U(s, a'; \theta)$, with $U(s, a'; \theta) = \alpha\ \mathbb{E}\left[Z(s, a'; \theta)\right]\ +\ (1 - \alpha)\ \text{VaR}_\rho\left[Z(s, a'; \theta)\right]$.
    Interact with the environment $\mathcal{E}$ with action a to get the next state s' and the reward r.
    Store the experience $e = (s, a, r, s')$ in the experience replay memory M.
    Randomly sample from M a minibatch of N_e experiences $e_i = (s_i, a_i, r_i, s_i^{'})$.
    Derive a discretisation of the domain $\mathcal{X}$ by sampling N_z returns $z \sim \mathcal{U}([z_{\min}, z_{\max}])$.
    for $i = 0$ **to** N_e **do**
      for all $z \in \mathcal{X}$ **do**
        if $s_i^{'}$ is terminal **then**
          Set $y_i(z) = \begin{cases} 0 & \text{if } z < r_i, \\ 1 & \text{otherwise.} \end{cases}$
        else
          Set $y_i(z) = Z\left(\frac{z - r_i}{\gamma} \middle| s_i^{'}, \text{argmax}_{a_i^{'} \in \mathcal{A}} U(s_i^{'}, a_i^{'}; \theta^-); \theta^-\right)$.
        end if
      end for
    end for
    Compute the loss $\mathcal{L}_C(\theta) = \sum_{i=0}^{N_e} \left(\sum_{z \in \mathcal{X}} \left(y_i(z) - Z(z|s_i, a_i; \theta)\right)^2\right)^{1/2}$.
    Clip the resulting gradient in the range $[0, 1]$.
    Update the main UMNN parameters θ using the ADAM optimiser with learning rate L_r.
    Update the target UMNN parameters $\theta^- = \theta$ every N^- steps.
    Anneal the ϵ-greedy exploration parameter ϵ.
  end for
end for

To conclude this section, ensuring the reproducibility of the results in a transparent way is particularly important to this research work. In order to achieve this, Table 5.1 provides a brief description of the key hyperparameters used in the experiments. Moreover, the entire experimental code is made publicly available at the following link:

`https://github.com/ThibautTheate/Risk-Sensitive-Policy-with-Distributional-Reinforcement-Learning`.

Table 5.1: Description of the main hyperparameters used in the experiments.

Hyperparameter	**Symbol**	**Value**
DNN structure	-	$[128, 128]$
Learning rate	L_r	10^{-4}
Deep learning optimiser epsilon	-	10^{-5}
Replay memory capacity	C	10^4
Batch size	N_e	32
Target update frequency	N^-	10^3
Random return resolution	N_z	200
Random return lower bound	$z_{\min}$	-2
Random return upper bound	$z_{\max}$	$+2$
Exploration ϵ-greedy initial value	-	1.0
Exploration ϵ-greedy final value	-	0.01
Exploration ϵ-greedy decay	-	10^4
Risk coefficient	ρ	10%
Risk trade-off	α	0.5

5.6 Results

The experiments conducted in the scope of this research work are subdivided into two parts. Firstly, the performance achieved by the resulting decision-making policies is quantitatively assessed in Section 5.6.1. Secondly, the probability distributions naturally learnt by the novel risk-sensitive distributional RL algorithm are analysed in Section 5.6.2.

5.6.1 Decision-making policy performance

To begin with, the performance achieved by the decision-making policies π learnt has to be evaluated, both in terms of expected outcome and risk. For comparison purposes, the results obtained by the well-established DQN algorithm, a reference without any form of risk sensitivity, are presented alongside those of the newly introduced RS-UMDQN-C algorithm. It shall also be mentioned that these two RL algorithms achieve very similar results when risk sensitivity is disabled ($\alpha = 1$ for the RS-UMDQN-C algorithm). Such a behaviour is expected and validates the risk-neutral version of the proposed methodology.

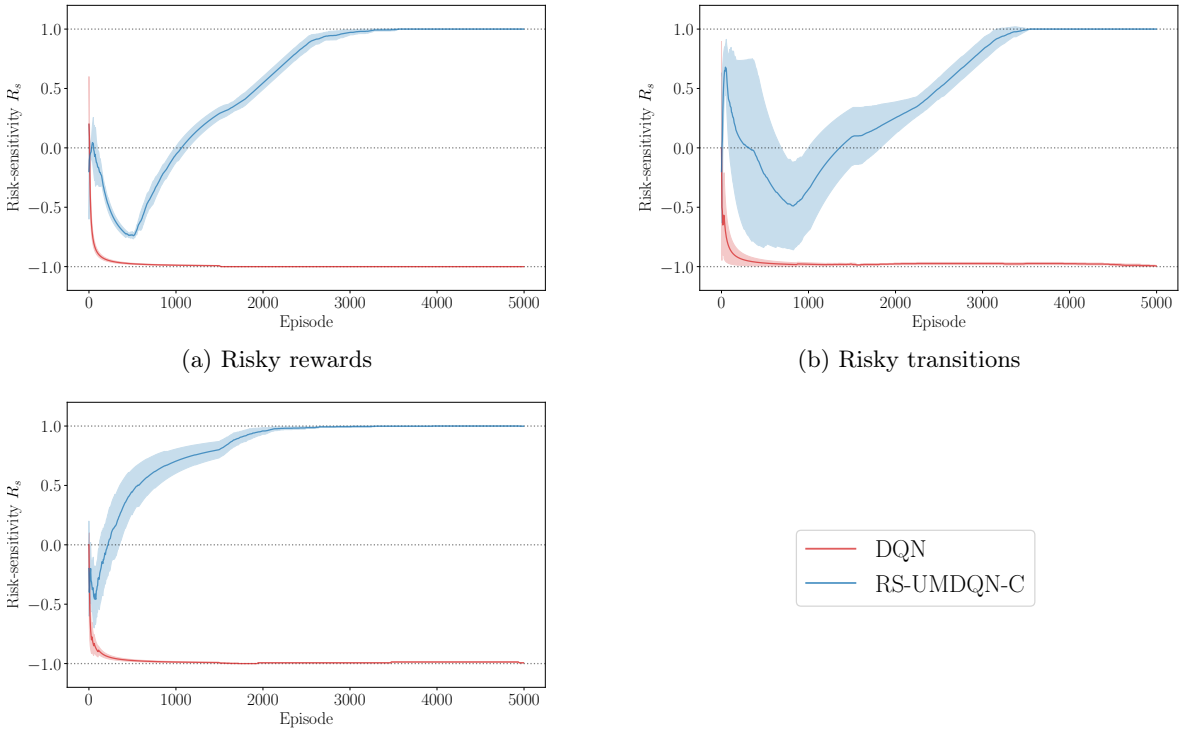
In the following, two analyses are presented. Firstly, the probability distribution of the (non discounted) cumulative reward of a learnt policy π, denoted $S^\pi \in \mathcal{Z}$, is investigated. More precisely, the expectation $\mathbb{E}[\cdot]$, the risk function $\mathcal{R}_\rho[\cdot]$ and the utility function $U[\cdot]$ of that random variable S^π are derived for each RL algorithm studied and compared. Secondly, this research work introduces a novel easy-to-interpret performance indicator $R_s \in [-1, 1]$ for evaluating the risk sensitivity of the decision-making policies learnt, by taking advantage of the simplicity of the benchmark environments presented in Section 5.5.1. In fact, it is made possible by the simple assessment from a human perspective of the relative riskiness of a path in the grid world environments studied. If the optimal path in terms of risk is chosen (green arrows in Figure 5.6), a positive score $R_s = +1$ is awarded. On the contrary, the riskier path but optimal in expectation (orange arrows in Figure 5.6) yields a negative score $R_s = -1$. If no objective nor trap areas are reached within the time allowed, a neutral score $R_s = 0$ is delivered. Therefore, the evolution of this performance indicator provides valuable information about the convergence of the RL algorithms towards the different possible paths as well as about the stability of the learning process. Formally, let $\tau = \{s_t, a_t\}_{t\in[0,T]}$ with $s_t \in \mathcal{S}$ and $a_t \in \mathcal{A}$ be a trajectory defined over a time horizon $T < 10$ (ending with a terminal state, and subject to an upper bound), and let τ_+ and τ_- respectively be the sets of trajectories associated to the green and orange paths in Figure 5.6. Based on these definitions, the risk-sensitivity R_s of a policy π is a random variable that can be assessed via Monte Carlo as the following:

$$R_s(\pi) = \begin{cases} +1 & \text{if } \pi \text{ produces trajectories } \{s_t, a_t\}_{t\in[0,T]} \in \tau_+ \text{ with } a_t = \pi(s_t), \\ -1 & \text{if } \pi \text{ produces trajectories } \{s_t, a_t\}_{t\in[0,T]} \in \tau_- \text{ with } a_t = \pi(s_t), \\ 0 & \text{otherwise.} \end{cases} \tag{5.22}$$

Following the methodology previously explained, the first results on policy performance are summarised in Table 5.2, which compares the decision-making policies learnt by both the DQN and RS-UMDQN-C algorithms, in terms of expected outcome and risk. The second results on policy performance are compiled in Figure 5.7, plotting the evolution of the risk-sensitivity performance indicator R_s during the training phase. It can be clearly observed from these two analyses that the proposed approach is effective in learning decision-making policies that are sensitive to the risk for relatively simple environments. Indeed, as expected, the DQN algorithm yields policies that are optimal in expectation whatever the level of risk incurred. In contrast, the RS-UMDQN-C algorithm is able to leverage both expected outcome and risk in order to learn decision-making policies that produce a slightly lower expected return with a significantly lower risk level. This allows the proposed methodology to significantly outperform the risk-neutral RL algorithm of reference with respect to the performance indicator of interest $U[S^\pi]$ in Table 5.2. Finally, it is also encouraging to observe from Figure 5.7 that the learning process seems to be quite stable for relatively simple environments, despite having to maximise a much more complicated function.

Table 5.2: Comparison of the expectation $\mathbb{E}[\cdot]$, risk function $\mathcal{R}_\rho[\cdot]$ and risk-based utility function $U[\cdot]$ of the cumulative reward S^π achieved by the decision-making policies π learnt by both the well-established DQN algorithm and the newly introduced RS-UMDQN-C algorithm.

Benchmark environment	**DQN**			**RS-UMDQN-C**		
	$\mathbb{E}[S^\pi]$	$\mathcal{R}_\rho[S^\pi]$	$U[S^\pi]$	$\mathbb{E}[S^\pi]$	$\mathcal{R}_\rho[S^\pi]$	$U[S^\pi]$
Risky rewards	**0.3**	-1.246	-0.474	0.1	**-0.126**	**-0.013**
Risky transitions	**0.703**	0.118	0.411	0.625	**0.346**	**0.485**
Risky grid world	**0.347**	-1.03	-0.342	0.333	**0.018**	**0.175**

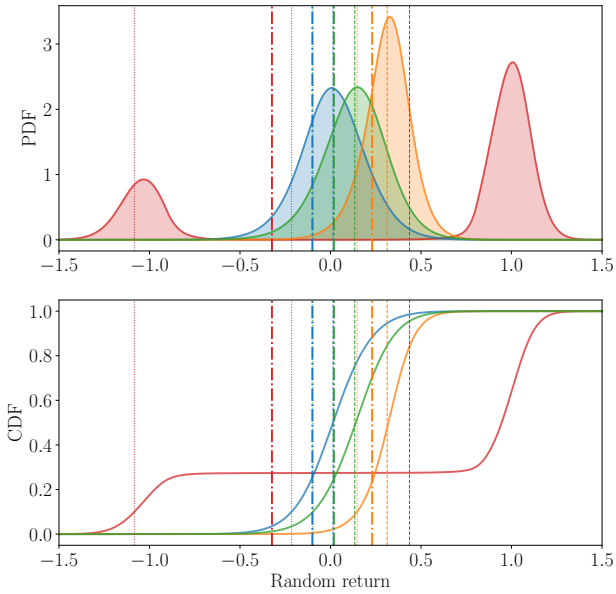


(a) Risky rewards

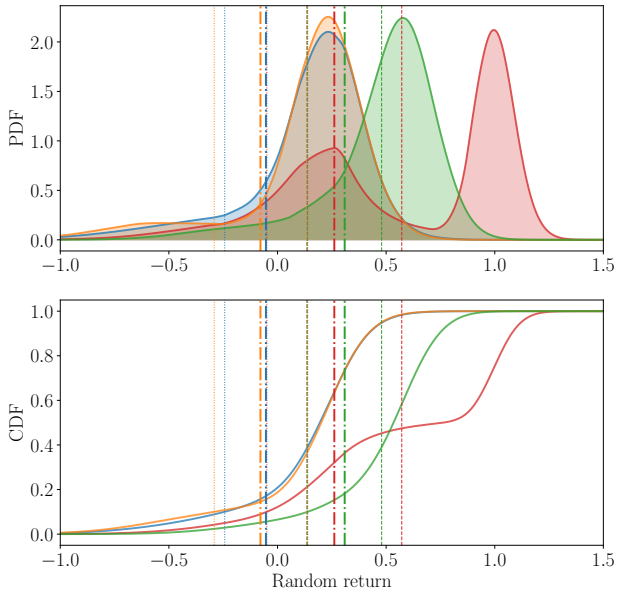
(b) Risky transitions

(c) Risky grid world

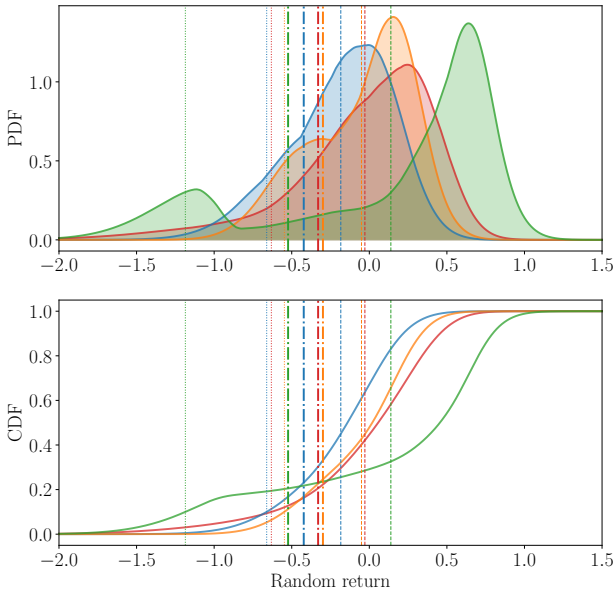
Figure 5.7: Evolution of the risk-sensitivity performance indicator R_s (expected value of the random variable) achieved by the decision-making policies π learnt by both the well-established DQN algorithm and the newly introduced RS-UMDQN-C algorithm during the training phase.



(a) Risky rewards



(b) Risky transitions



(c) Risky grid world

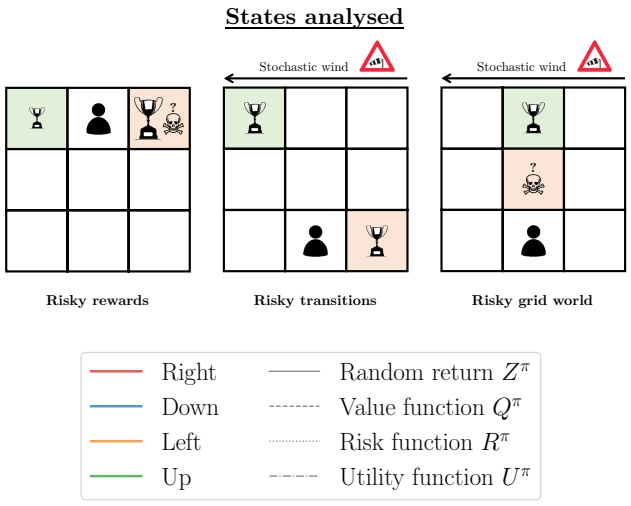


Figure 5.8: Visualisation of the random return probability distributions Z^π naturally learnt by the newly introduced RS-UMDQN-C algorithm for typical states of the benchmark environments, together with the value, risk and utility functions derived (Q^π, R^π and U^π).

5.6.2 Probability distribution visualisation

As previously claimed, an important advantage of the proposed solution is the enhanced interpretability of the decision-making process. Indeed, understanding and motivating the decisions outputted by the resulting policy π is greatly facilitated by having access to the probability distributions of the random return jointly learnt. In addition, the analysis and comparison of the value, risk and utility functions (Q^π, R^π and U^π) associated with different actions provide a valuable summary about the decision-making process, but also about the control problem itself. Moreover, such an analysis may be particularly important to correctly tune the risk trade-off parameter α according to the user's risk aversion.

As an illustration, Figure 5.8 displays some random return probability distributions Z^π that are learnt by the RS-UMDQN-C algorithm jointly to the risk-sensitive policy π. More precisely, a single relevant state is selected for analysis for each benchmark environment. This choice is based on the importance of the next decision in following a clear path, either maximising the expected outcome or mitigating the risk. Firstly, it can be observed that the risk-sensitive distributional RL algorithm does manage to accurately learn the probability distribution of the random return, qualitatively from a human perspective. In particular, the multimodality purposely designed to create risky situations appears to be well preserved. Such a result is particularly encouraging since this feature is essential to the success of the proposed solution, by ensuring the accurate estimation of the risk as defined in Section 5.4.2. This observation is in line with the findings of Chapter 4 introducing the UMDQN algorithm, and suggests that the solution introduced to achieve risk-sensitivity does not significantly alter the properties of the original distributional RL algorithm. Secondly, as previously explained, Figure 5.8 highlights the relevance of each function introduced (Q^π, R^π and U^π) for making and motivating a decision made by a RL algorithm. Their analysis truly contributes to the understanding of the potential trade-off between expected performance maximisation and risk mitigation for a given decision-making problem, as well as the extent to which different values of the important parameter α leads to divergent policies.

5.7 Conclusion

This doctoral thesis chapter introduces a straightforward yet efficient solution to learning risk-sensitive decision-making policies on the basis of the distributional RL approach. The proposed methodology presents key advantages. Firstly, it is perfectly compatible with any distributional RL algorithm, and requires minimal modification to the original algorithm. Secondly, the simplicity of the approach truly contributes to the interpretability and ease of analysis of the resulting risk-sensitive policies, a particularly important feature to avoid black-box machine learning models. Lastly, the solution presented enables to cover the complete potential trade-off between expected outcome maximisation and risk minimisation. The first experiments conducted on three relevant toy problems yield promising results, which may be viewed as a proof of concept for the accessible and practical solution introduced.

To conclude this thesis chapter, some interesting leads can be suggested as future work. Firstly, the research conducted is exclusively empirical and does not study any theoretical guarantees about the resulting risk-sensitive distributional RL algorithms. Among others, the study of the convergence of these algorithms would be a relevant future research direction. Secondly, building on the promising results achieved, the solution presented should definitely be evaluated on more complex environments, for which the risk should ideally be mitigated. Lastly, the approach could be extended to not only mitigate the risk but also to completely discard actions that would induce an excessive level of risk, in order to increase compliance with the objective criterion originally defined in Section 5.4.1 for risk-sensitive RL.

The important thing is to never stop questioning.

— Albert Einstein

Chapter 6

Matching of Everyday Power Supply and Demand with Dynamic Pricing: Problem Formalisation and Conceptual Analysis

Figure 6.1: Illustration of Chapter 6 entitled *Matching of Everyday Power Supply and Demand with Dynamic Pricing: Problem Formalisation and Conceptual Analysis*, created by a generative art AI [1].

Chapter overview

The ongoing energy transition is expected to significantly increase the share of renewable energy sources whose production is intermittent in the electricity mix. Apart from key benefits, this development has the major drawback of generating a mismatch between power supply and demand. The innovative *dynamic pricing* approach may contribute to mitigating that critical problem by taking advantage of the flexibility offered by the demand side. At its core, this idea consists in providing the consumer with a price signal which is evolving over time, in order to influence its consumption. This novel approach involves a challenging decision-making problem that can be summarised as follows: how to determine a price signal maximising the synchronisation between power supply and demand under the complex constraints of maintaining the producer/retailer's profitability while also benefiting the final consumer at the same time? As a contribution, this thesis chapter presents a detailed formalisation of this particular decision-making problem. Moreover, the research work discusses the diverse algorithmic components necessary to efficiently design a dynamic pricing policy: different forecasting models together with an accurate statistical modelling of the demand response to dynamic prices.

This thesis chapter is primarily based on the following scientific publication [6]:

Thibaut Théate, Antonio Sutera, and Damien Ernst. Matching of everyday power supply and demand with dynamic pricing: Problem formalisation and conceptual analysis. *Energy Reports*, 9:2453–2462, 2023.

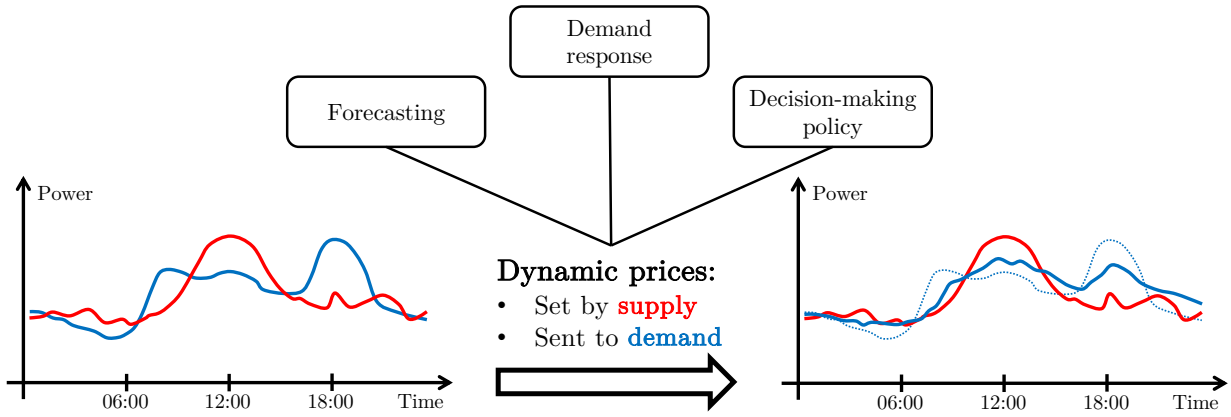


Figure 6.2: General illustration of the methodology presented in this thesis chapter entitled *Matching of Everyday Power Supply and Demand with Dynamic Pricing: Problem Formalisation and Conceptual Analysis*.

6.1 Introduction

Climate change is undeniably a major challenge facing humanity in the 21st century [114]. An ambitious transformation is required in all sectors to significantly lower their respective carbon footprints. Electricity generation is no exception, with the burning of fossil fuels, mainly coal and gas, being by far the dominant power source in the world today [115]. This sector has to undergo an important transformation of the global electricity mix by promoting power sources with a significantly lower carbon footprint. Belonging to that category are nuclear power, hydroelectricity, biomass or geothermal energy which are relatively controllable, but also the energy directly extracted from wind and sun which is conversely intermittent in nature. Since wind turbines and photovoltaic panels are expected to play a key role in the energy transition, solutions are required to address their variable production. Interesting technical avenues are the interconnection of power grids [116] and the development of storage capacities such as batteries, pumped hydroelectricity or hydrogen [117]. Another promising and innovative solution is to rationally influence the behaviour of consumers through the use of *dynamic pricing* (DP), so that power supply and demand are better synchronised. In fact, the core idea is to take advantage of the flexibility offered by the power demand side.

The dynamic pricing approach consists in continuously adapting the electricity price that the final consumer has to pay in order to influence its consumption behaviour. Basically, when demand exceeds supply, the price would be increased in order to take down consumption. Conversely, a reduced power price would be provided when there is excessive production compared to consumption. From a graphical perspective, the objective is not only to shift the daily consumption curve but also to change its shape in order to better overlap with the intermittent production curve of renewable energy sources, as illustrated in Figure 6.3.

The innovative dynamic pricing approach relies on two important assumptions. Firstly, the final consumer has to be equipped with a smart metering device to measure its production in real-time and with communication means for the price signal. Secondly, the final consumer has to be able to provide a certain amount of flexibility regarding its power consumption. Moreover, it has to be sufficiently receptive to the incentives offered to reduce its electricity bill in exchange for a behaviour change. If these important requirements are met, the major strength of the dynamic pricing approach is its potential benefits for both the consumer and the producer/retailer. Moreover, these benefits would not only be in terms of economy, but also potentially in terms of ecology and autonomy. In fact, dynamic prices may be seen as a mechanism rewarding the flexibility of the demand side.

The contributions of this research work are twofold. Firstly, the complex decision-making problem faced by a producer/retailer willing to develop a dynamic pricing strategy is presented and rigorously formalised. Secondly, the diverse algorithmic components required to efficiently design a dynamic pricing policy are thoroughly discussed. To the authors' knowledge, demand response via dynamic pricing has received considerable attention from the research community, but from the perspective of the demand side alone. Therefore, the present research may be considered as a pioneer work studying dynamic pricing from the perspective of the supply side for taking advantage of the flexibility of the power consumers.

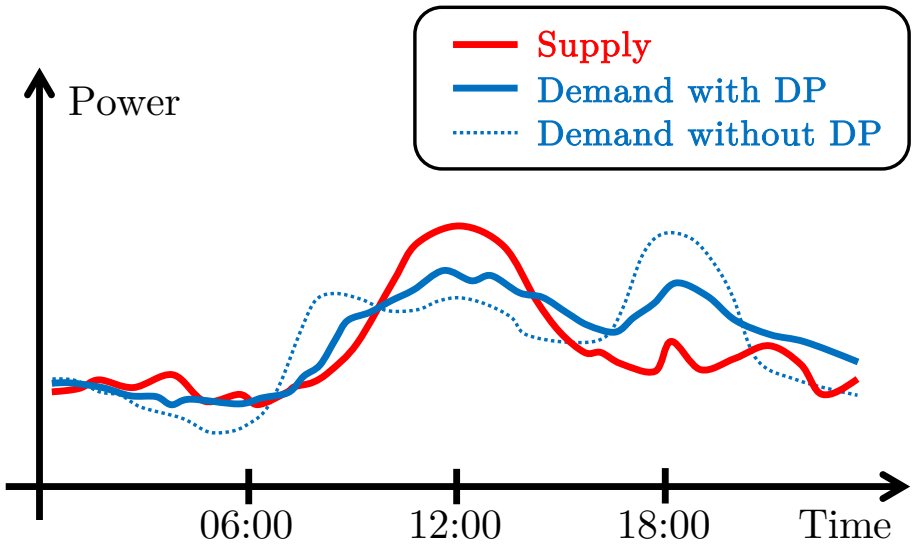


Figure 6.3: Illustration of the dynamic pricing approach's potential to shift and change the shape of a typical daily consumption curve (blue) so that there is a better synchronisation with the daily intermittent production of renewable energy sources (red).

6.2 Literature review

Over the last decade, the management of the power demand side in the scope of the energy transition has received increasing attention from the research community. In fact, there exist multiple interesting generic approaches when it comes to demand response. Without getting into too many details, the scientific literature includes several surveys summarising and discussing the different techniques available together with their associated challenges and benefits [118, 119, 120, 121, 122]. In this research work, the focus is exclusively set on the demand response induced by dynamic electricity prices.

As previously mentioned, the scientific literature about demand response via dynamic pricing is primarily focused on the perspective of the demand side. Multiple techniques have already been proposed to help the consumer provide flexibility and take advantage of small behavioural changes to lower its electricity bill. For instance, [123] presents a power scheduling method based on a genetic algorithm to optimise residential demand response via an energy management system, so that the electricity cost is reduced. In [124], a technique based on dynamic programming is introduced for determining the optimal schedule of residential controllable appliances in the context of time-varying power pricing. One can also mention [125] that proposes an energy sharing model with price-based demand response for microgrids of peer-to-peer prosumers. The approach is based on a distributed iterative algorithm and has been shown to lower the prosumers' costs as well as to improve the sharing of photovoltaic energy. More recently, (deep) reinforcement learning techniques have been proven to be particularly relevant for controlling the residential demand response in the particular context of dynamic power prices [126, 127].

On the contrary, the question of inducing a relevant residential demand response based on a dynamic pricing approach from the perspective of the supply side has not received a lot of attention from the research community yet. Still, there are a few works in the scientific literature about the mathematical modelling of the demand response caused by dynamic power prices, which is a key element in achieving the former objective. To begin with, [128] presents a simulation model highlighting the evolution of electricity consumption profiles when shifting from a fixed tariff to dynamic power prices. A similar goal is pursued by [129] which introduces a fully data-driven approach relying on the data collected by smart meters and exogenous variables. The resulting simulation model is based on consumption profiles clustering and conditional variational autoencoders. Alternatively, [130] presents a functional model of residential power consumption elasticity under dynamic pricing to assess the impact of different electricity price levels, based on a Bayesian probabilistic approach. In addition to these mathematical models, one can also mention some real-life experiments conducted to assess the responsiveness of the residential electricity demand to dynamic pricing [131, 132].

6.3 Problem formalisation

This section presents a mathematical formalisation of the challenging sequential decision-making problem related to the dynamic pricing approach for inducing a relevant residential demand response. To begin with, the contextualisation considered for studying this particular problem is briefly described, followed by an overview of the decision-making process. Then, a discretisation of the continuous timeline is introduced. Subsequently, the formal definition of a dynamic pricing policy is presented. Lastly, the input and output spaces of a dynamic pricing policy are described, together with the objective criterion.

6.3.1 Contextualisation

As previously hinted, the present research work focuses on the interesting real-case scenario of a producer/retailer whose production portfolio is composed of an important share of renewable energy sources such as wind turbines and photovoltaic panels. Because of the substantial intermittency of these generation assets, a strong connection to the different energy markets is required in order to fully satisfy its customers regardless of the weather. Nevertheless, the consumers are assumed to be well informed and willing to adapt their behaviour in order to consume renewable energy rather than electricity purchased on the market whose origin may potentially be unknown. Within this particular context, the benefits of the dynamic pricing approach taking advantage of the consumers' flexibility are maximised. Indeed, the insignificant marginal cost associated with these intermittent renewable energy sources coupled with their low carbon footprint make this innovative approach interesting from an economical perspective for both supply and demand sides, but also in terms of ecology. Moreover, the autonomy of the producer/retailer is expected to be reinforced by lowering its dependence on the energy markets. Concurrently, the problematic dependence on fossil fuels such as coal and gas will be reduced as well.

In this research work, the predicted difference between power production and consumption is assumed to be fully secured in the day-ahead electricity market. Also called spot market, the day-ahead market has an hourly resolution and is operated once a day for all hours of the following day via a single-blind auction. In other words, trading power for hour H of day D has to be performed ahead on day $D-1$ between 00:00 AM (market opening) and 12:00 AM (market closure). Therefore, assuming that the trading activity occurs just before market closure, the energy is at best purchased 12 hours (for delivery at 00:00 AM of day D) up to 35 hours (for delivery at 11:00 PM of day D) before the actual delivery of power. Apart from the day-ahead electricity market, it is assumed that there are no trading activities on the future/forward nor intraday markets. Nevertheless, if there remains an eventual mismatch between production and consumption at the time of power delivery, the producer/retailer would be exposed to the imbalance market. In this case, the so-called imbalance price has to be inevitably paid as compensation for pushing the power grid off balance.

6.3.2 Decision-making process overview

The decision-making problem studied in this research work is characterised by a particularity: a variable time lag between the moment a decision is made and when it becomes effective. As previously hinted, any remaining difference between production and consumption after demand response has to ideally be traded on the day-ahead market. The purpose of this assumption is to limit the exposure of the producer/retailer to the imbalance market. For this reason, the complete price signal sent to the consumer on day D has to be generated before the closing of the day-ahead market at midday on day $D-1$. Additionally, it is assumed that the price signal cannot be refreshed afterwards.

Basically, the decision-making problem at hand can be formalised as follows. The core objective is to determine a decision-making policy, denoted Π, mapping at time τ input information of diverse nature I_τ to the electricity price signal S_τ to be sent to the consumers over a future time horizon well-defined:

$$S_\tau = \Pi(I_\tau), \tag{6.1}$$

where:

- I_τ represents the information vector gathering all the available information of diverse nature at time τ which may be helpful to make a relevant dynamic pricing decision,
- S_τ represents a series of electricity prices generated at time τ and shaping the dynamic price signal over a well-defined future time horizon.

In fact, the dynamic pricing approach from the perspective of the supply side belongs to a particular class of decision-making problems: automated planning and scheduling. Contrarily to conventional decision-making outputting one action at a time, planning decision-making is concerned with the generation of a sequence of actions. In other words, a planning decision-making problem requires to synthesise in advance a strategy or plan of actions over a certain time horizon. Formally, the decision-making has to be performed at a specific time τ about

a control variable over a future time horizon beginning at time $\tau_i > \tau$ and ending at time $\tau_f > \tau_i$. In this case, the decision-making is assumed to be performed on day $D-1$ just before the closing of the day-ahead market at 12:00 AM to determine the price signal to be sent to the consumers throughout the entire following day D (from 00:00 AM to 11:59 PM).

In the next sections, a more accurate and thorough mathematical formalisation of the dynamic pricing problem from the perspective of the supply side is presented. Moreover, the planning problem previously introduced is cast into a sequential decision-making problem. Indeed, this research work intends to focus on a decision-making policy outputting a single price from the signal S_τ at a time based on a subset of the information vector I_τ. Such an important choice naturally comes with both its advantages and limitations.

6.3.3 Timeline discretisation

Theoretically, the dynamic power price signal sent to the consumer could be continuously changing over time. More realistically, the present research work adopts a discretisation of the continuous timeline so that the electricity price is adapted at regular intervals. Formally, this timeline is discretised into a number of time steps t spaced by a constant duration Δt. If the duration Δt is too large, the synchronisation improvement between supply and demand will probably be of poor quality. Conversely, lowering the value of the duration Δt increases the complexity of the decision-making process, and a too high update frequency may even confuse the consumer. There is a trade-off to be found concerning this important parameter. In this research work, the dynamic price signal is assumed to change once per hour, meaning that Δt is equal to one hour. This choice is motivated by the hourly resolution of the day-ahead market, which has proven to be an appropriate compromise over the years for matching power production and consumption. Another relevant discretisation choice could be to have a price signal which is updated every quarter of an hour. In the rest of this research work, the increment (decrement) operations $t+1$ $(t-1)$ are used to model the discrete transition from time step t to time step $t+\Delta t$ $(t-\Delta t)$, for the sake of clarity.

6.3.4 Dynamic pricing policy

Within the context previously described, a dynamic pricing planning policy Π consists of the set of rules adopted to make a decision regarding the future price signal sent to the power consumers over the next day. In fact, this research work makes the choice to decompose this planning policy into a set of 24 dynamic pricing decision-making policies π, each one outputting a single electricity price for one hour of the following day. Mathematically, such a dynamic pricing strategy can be defined as a programmed policy $\pi : \mathcal{X} \rightarrow \mathcal{Y}$, either deterministic or stochastic, which outputs a decision $y_t \in Y$ for time step t based on some input information $x_t \in \mathcal{X}$ so as to maximise an objective criterion. The input x_t can be derived from the information vector I_τ associated with the decision-making for time step t, after some potential preprocessing operations. Concerning the output, the price signal S_τ is composed of a sequence of 24 dynamic pricing policy outputs y_t.

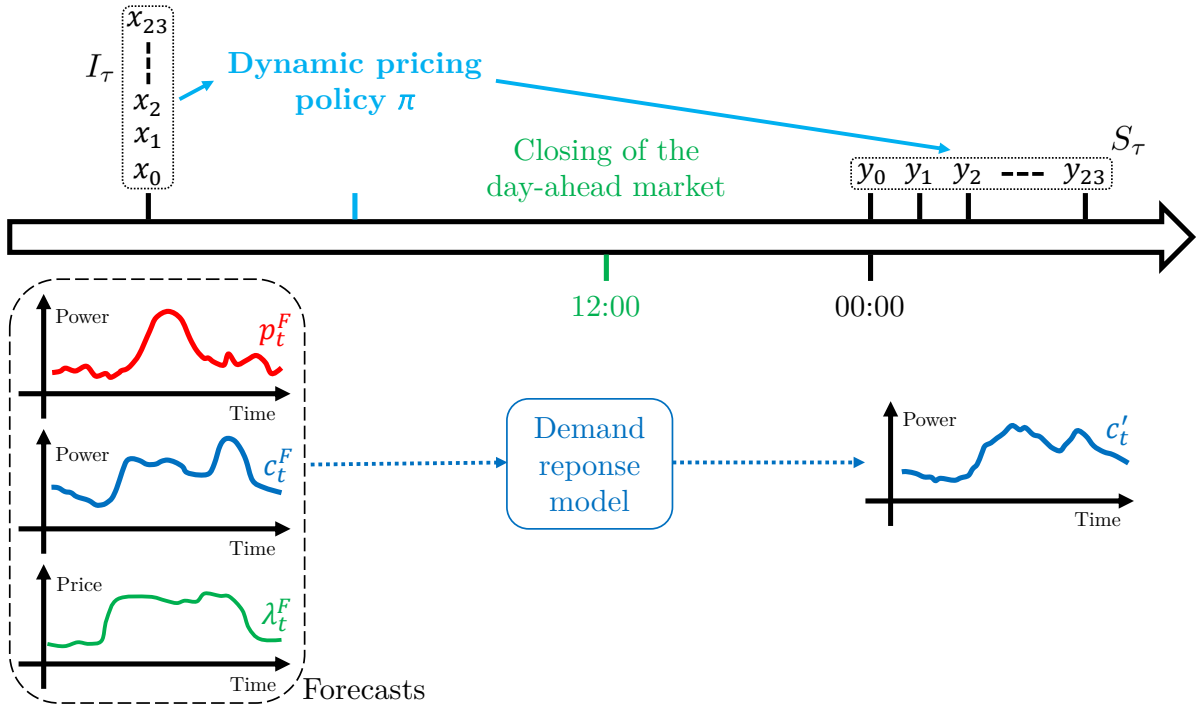


Figure 6.4: Illustration of the formalised decision-making problem related to dynamic pricing from the perspective of the supply side. The notations x_t and y_t represent the inputs and outputs of a dynamic pricing policy π, which are not shown concurrent on the timeline since the decision-making occurs hours before the application of the dynamic pricing signal. The time axis of the four plots represents the complete following day for which the dynamic prices are generated. The mathematical notations p_t^F, c_t^F and λ_t^F respectively represent the forecast production, consumption and day-ahead market price for time step t. The quantity c_t' is the predicted consumption at time step t after taking into account the dynamic pricing signal.

In the rest of this research work, the time at which the decision-making does occur should not be confused with the time at which the dynamic price signal is active, meaning charging for energy consumption. The proposed formalisation assumes that the time step t refers to the time at which the dynamic price is active, not decided. Therefore, the decision-making of the dynamic pricing policy π for time step t ($y_t = \pi(x_t)$) is in fact performed hours in advance of time step t. This particularity is illustrated in Figure 6.4 describing the formalised decision-making problem related to dynamic pricing from the perspective of the supply side.

6.3.5 Input of a dynamic pricing policy

The input space $\mathcal{X}$ of a dynamic pricing policy π comprises all the available information which may help to make a relevant decision about future electricity prices so that an appropriate demand response is induced. Since the decision-making occurs 12 up to 35 hours in advance of the price signal activation, this information mainly consists of forecasts and estimations that are subject to uncertainty. As depicted in Figure 6.4, the dynamic pricing policy input $x_t \in \mathcal{X}$ refers to the decision-making occurring at time $\tau = t - h$ with $h \in [12,\ 35]$ about the dynamic pricing signal delivered to the consumer at time step t. In fact, the quantity I_τ may be seen as the information contained in the 24 inputs x_t for $t \in \{\tau + 12, ..., \tau + 35\}$.

Formally, the input $x_t \in \mathcal{X}$ is decided to be defined as the following:

$$x_t = \{P_t^F,\ C_t^F,\ \Lambda_t^F,\ Y_t,\ \mathcal{M}\}, \tag{6.2}$$

where:

- $P_t^F = \{p_{t+\epsilon}^F \in \mathbb{R}^+ \mid \epsilon = -k, ..., k\}$ represents a set of forecasts for the power production within a time window centred around time step t and of size k,
- $C_t^F = \{c_{t+\epsilon}^F \in \mathbb{R}^+ \mid \epsilon = -k, ..., k\}$ represents a set of forecasts for the power consumption within a time window centred around time step t and of size k,
- $\Lambda_t^F = \{\lambda_{t+\epsilon}^F \in \mathbb{R} \mid \epsilon = -k, ..., k\}$ represents a series of forecasts for the day-ahead market prices within a window centred around time step t and of size k,
- $Y_t = \{y_{t-\epsilon} \in \mathbb{R} \mid \epsilon = 1, ..., k\}$ represents the series of k previous values for the dynamic price signal sent to the final consumer,
- $\mathcal{M}$ is a mathematical model of the demand response that can be expected from the consumption portfolio, together with the required input information associated.

6.3.6 Output of a dynamic pricing policy

The output space $\mathcal{Y}$ of a dynamic pricing policy π solely includes the future price signal to be sent to the consumer. Formally, the dynamic pricing policy output $y_t \in \mathcal{Y}$, which represents the electricity price to be paid by the consumer for its power consumption at time step t, can be mathematically defined as follows:

$$y_t = e_t, \tag{6.3}$$

where $e_t \in \mathbb{R}$ represents the dynamic electricity price to be paid by the demand side for its power consumption at time step t. Out of the scope of this research work is the presentation of this price signal so that the impact on the final consumer is maximised. Indeed, to achieve better results, the way of communicating the output of the dynamic pricing policy has to be adapted to the audience, be it humans with different levels of electricity market expertise or algorithms (energy management systems).

6.3.7 Objective criterion

The dynamic pricing approach can provide multiple benefits, in terms of economy, ecology but also autonomy. Consequently, the objective criterion to be maximised by a dynamic pricing policy π is not trivially determined. In fact, several core objectives can be identified:

- maximising the match between supply and demand,
- minimising the carbon footprint of power generation,
- minimising the electricity costs for the consumer,
- maximising the revenue of the producer/retailer.

Although some objectives do overlap, these four criteria are not completely compatible. For instance, maximising the synchronisation between power supply and demand may be equivalent to minimising the carbon footprint associated with the generation of electricity. Indeed, the production portfolio of the producer/retailer being mainly composed of intermittent renewable energy sources, its energy is likely to have a reduced carbon footprint compared to the electricity that can be purchased on the day-ahead market whose origin is unknown. On the contrary, maximising the revenue of the producer/retailer will obviously not lead to a minimised electricity bill for the consumer. This research work makes the choice to prioritise the maximisation of the synchronisation between supply and demand, and equivalently the minimisation of the carbon footprint, while translating the other two core objectives into relevant constraints. Firstly, the power costs for the consumer have to be reduced with respect to the situation without dynamic pricing. Secondly, the profitability of the producer/retailer has to be guaranteed.

Formally, the objective criterion to be optimised by a dynamic pricing policy π can be mathematically defined as the following. First of all, the main target to evaluate is the synchronisation between supply and demand, which can be quantitatively assessed through the deviation Δ_T. This quantity has to ideally be minimised, and can be mathematically expressed as follows:

$$\Delta_T = \sum_{t=0}^{T-1} |p_t - c_t|, \tag{6.4}$$

where:

- $t = 0$ corresponds to the first electricity delivery hour of a new day (00:00 AM),
- T is the time horizon considered, which should be a multiple of 24 to have full days,
- p_t is the actual power production (not predicted) from the supply side at time step t,
- c_t is the actual power consumption (not predicted) from the demand at time step t.

Afterwards, the first constraint concerning the reduced power costs for the consumer has to be modelled mathematically. This is achieved via the electricity bill B_T paid by the consumer over the time horizon T, which can be expressed as the following:

$$B_T = \sum_{t=0}^{T-1} c_t \; y_t \; . \tag{6.5}$$

As previously explained, the consumer power bill B_T should not exceed that obtained without dynamic pricing. In that simpler case, the consumer is assumed to pay a price $\overline{e_t}$, which can for instance be a fixed tariff or a price indexed on the day-ahead market price. The situation without dynamic pricing is discussed in more details in Section 6.5. Consequently, the first constraint can be mathematically expressed as follows:

$$\sum_{t=0}^{T-1} c_t \; y_t \leq \sum_{t=0}^{T-1} \overline{c_t} \; \overline{e_t} \; , \tag{6.6}$$

where $\overline{c_t}$ is the actual electrical power consumption (not predicted) from the demand side at time step t without any dynamic pricing mechanism.

Then, the second constraint is about the profitability of the producer/retailer, which is naturally achieved if its revenue exceeds its costs. The revenue R_T of the producer/retailer over the time horizon T can be mathematically expressed as the following:

$$R_T = \sum_{t=0}^{T-1} \left[c_t \; y_t - (c'_t - p_t^F) \; \lambda_t - (c_t - p_t) \; i_t \right], \tag{6.7}$$

where:

- λ_t is the actual power price (not predicted) on the day-ahead market at time step t,
- i_t is the actual imbalance price (not predicted) on the imbalance market at time step t,
- c'_t is the predicted power consumption at time step t after the demand response to the dynamic prices, based on the demand response mathematical model $\mathcal{M}$.

The first term simply corresponds to the payment of the customers for their electricity consumption. The second term can either be a revenue or a cost induced by the predicted mismatch between supply and demand, which is traded on the day-ahead market. The last term is the cost or revenue caused by the remaining imbalance between supply and demand, which has to be compensated in the imbalance market.

The total costs incurred by the producer/retailer at each time step t can be decomposed into both fixed costs F_C and marginal costs M_C. In this particular case, the marginal costs of production are assumed to be negligible since the production portfolio is composed of intermittent renewable energy sources such as wind turbines and photovoltaic panels. Therefore, the second constraint can be mathematically expressed as follows:

$$\sum_{t=0}^{T-1} \left[c_t \; y_t - (c'_t - p_t^F) \; \lambda_t - (c_t - p_t) \; i_t \right] \geq F_C \; T \; . \tag{6.8}$$

Finally, the complete objective criterion to be optimised by a dynamic pricing policy π can be mathematically expressed as the following:

$$\begin{aligned} \underset{\pi}{\text{minimise}} \quad & \sum_{t=0}^{T-1} |p_t - c_t|, \\ \text{subject to} \quad & R_T \geq F_C \; T \; , \\ & B_T \leq \sum_{t=0}^{T-1} \overline{c_t} \; \overline{e_t} \; . \end{aligned} \tag{6.9}$$

6.4 Algorithmic components discussion

This section presents a thorough discussion about the different algorithmic modules required to efficiently design a dynamic pricing policy from the perspective of the supply side. Firstly, the three necessary forecasting blocks are rigorously analysed. Secondly, the mathematical modelling of the demand response induced by dynamic power prices is discussed. Lastly, the proper management of uncertainty is considered.

In parallel, for the sake of clarity, Figure 6.5 highlights the interconnections between the different algorithmic components in the scope of a dynamic pricing policy from the perspective of the supply side. Moreover, Algorithm 9 provides a thorough description of the complete decision-making process for the dynamic pricing problem at hand. The complexity of the variable time lag between decision-making and application is highlighted. Assuming that the decision-making occurs once a day at 12:00 AM just before the closing of the day-ahead market for all hours of the following day, the dynamic price at time step t is decided hours in advance at time step $t-[12+(t\%24)]$, with the symbol % representing the modulo operation.

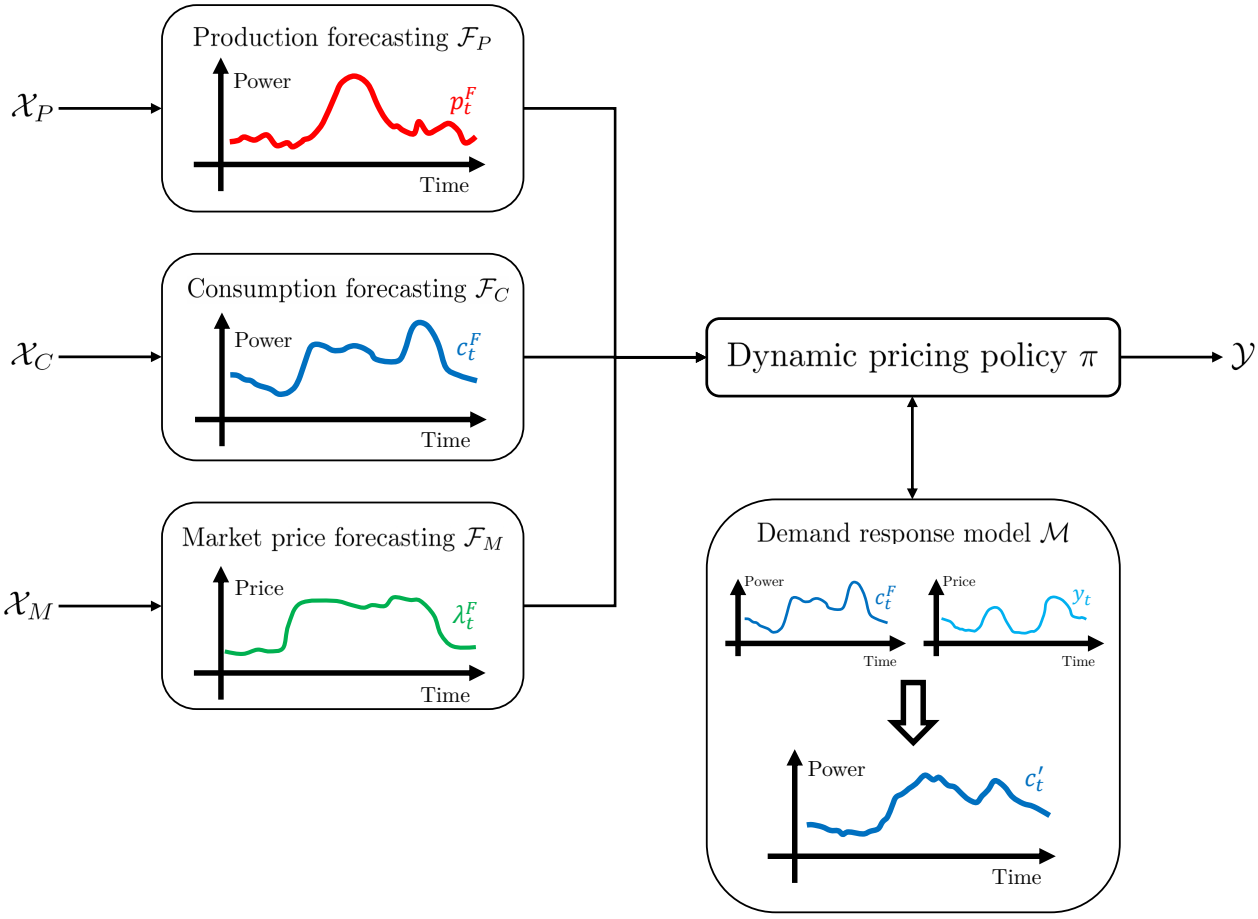


Figure 6.5: Illustration of the decision-making process related to dynamic pricing from the perspective of the supply side, with the connections between the different algorithmic components highlighted.

Algorithm 9 Dynamic pricing complete decision-making process

The decision-making occurs once per day just before the closing of the day-ahead market at 12:00 AM of day $D-1$ for all hours of the following day D.
The decision-making for the dynamic price of time step t actually occurs in advance at time step $t-[12+(t\%24)]$.
for $\tau = -12$ **to** $T-12$ **do**
  Check whether the time is 12:00 AM to proceed to the decision-making phase.
  if $(\tau+12)\%24 = 0$ **then**
    for $t = \tau+12$ **to** $\tau+35$ **do**
      Gather the available information for production forecasting $x_t^P = \{W_t^F,\ A_t^F,\ I_t^P\}$.
      Gather the available information for consumption forecasting $x_t^C = \{W_t^F,\ T_t,\ I_t^C\}$.
      Gather the available information for day-ahead market price forecasting $x_t^M = \{x_t^P,\ x_t^C,\ G_t^F,\ M_t,\ I_t^M\}$.
      Forecast production at time step t: $p_t^F = \mathcal{F}_P\left(x_t^P\right)$.
      Forecast consumption at time step t: $c_t^F = \mathcal{F}_C\left(x_t^C\right)$.
      Forecast the day-ahead market price at time step t: $\lambda_t^F = \mathcal{F}_M\left(x_t^M\right)$.
    end for
    for $t = \tau+12$ **to** $\tau+35$ **do**
      Gather the input information for the dynamic pricing policy $x_t = \{P_t^F,\ C_t^F,\ \Lambda_t^F,\ Y_t,\ \mathcal{M}\}$.
      Make a dynamic pricing decision for time step t: $y_t = \pi(x_t)$.
    end for
    Announce the dynamic prices for all hours of the following day $\{y_t \mid t = \tau+12, ..., \tau+35\}$.
  end if
end for

6.4.1 Power production forecasting

The first forecasting block to be discussed concerns the production of intermittent renewable energy sources such as wind turbines and photovoltaic panels. Having access to accurate predictions about the future output of the production portfolio is key to the performance of a dynamic pricing policy from the perspective of the supply side. As previously explained in Section 6.3, the forecasts have to be available one day ahead before the closing of the day-ahead electricity market for all hours of the following day. Naturally, the generation of such predictions introduces uncertainty, a complexity that has to be seriously taken into consideration in order to design sound dynamic pricing policies.

Formally, the forecasting model associated with the output of the production portfolio is denoted $\mathcal{F}_P$. Its input space $\mathcal{X}_P$ comprises every piece of information that may potentially have an impact on the generation of electricity from intermittent renewable energy sources such as wind turbines and photovoltaic panels for a particular time period of interest. Its output space $\mathcal{Y}_P$ is composed of a forecast regarding the power generation from the production portfolio for that same time period. Mathematically, the forecasting model input $x_t^P \in \mathcal{X}_P$ and output $y_t^P \in \mathcal{Y}_P$ at time step t can be expressed as follows:

$$x_t^P = \{W_t^F,\ A_t^F,\ I_t^P\}, \tag{6.10}$$

$$y_t^P = p_t^F, \tag{6.11}$$

where:

- W_t^F represents various weather forecasts that are related to the power production of intermittent renewable energy sources such as wind turbines and photovoltaic panels (wind speed and direction, solar irradiance, etc.) at the time step t,
- A_t^F represents predictions about the available capacity of the production portfolio at time step t, that is impacted by scheduled maintenance, repairs, or other similar constraints,
- I_t^P represents any additional information that may help to accurately forecast the future power generation of the producer/retailer's production portfolio at time step t.

In the scientific literature, the current state-of-the-art approach for forecasting the power production of intermittent renewable energy sources is primarily based on deep learning techniques together with some data cleansing processes and data augmentation approaches. The best architectures are *recurrent neural networks* (RNN), *convolutional neural networks* (CNN) and *transformers* [133, 134, 135, 136, 137].

6.4.2 Power consumption forecasting

The objective of the next important forecasting model deserving a discussion is to accurately predict the future power demand of the consumption portfolio before any demand response phenomenon is induced. Since the main goal of a dynamic pricing policy is to maximise the synchronisation between supply and demand, power load forecasts are of equal importance to electricity generation predictions. Similarly to the latter, the portfolio consumption forecasts are assumed to be generated one day ahead before the closing of the day-ahead market for all 24 hours of the following day. Additionally, the uncertainty associated with these predictions has to be carefully taken into account for the success of the dynamic pricing policy.

From a more formal perspective, the forecasting model responsible for predicting the future electricity load of the consumption portfolio is denoted $\mathcal{F}_C$. Its input space $\mathcal{X}_C$ includes all the information that may have an influence on the residential electricity consumption for a time period of interest. Its output space $\mathcal{Y}_C$ comprises a forecast of the power used by the consumption portfolio for that same time period. Mathematically, the consumption forecasting model input $x_t^C \in \mathcal{X}_C$ and output $y_t^C \in \mathcal{Y}_C$ at time step t are expressed as follows:

$$x_t^C = \{W_t^F,\ T_t,\ I_t^C\}, \tag{6.12}$$

$$y_t^C = c_t^F, \tag{6.13}$$

where:

- W_t^F represents various weather forecasts that are related to the residential electricity consumption (temperature, hygrometry, etc.) at the time step t,
- T_t represents characteristics related to the time step t (hour, weekend, season, holiday),
- I_t^C represents supplementary information that could potentially have an influence on the residential power consumption at time step t.

Similarly to renewable energy production forecasting, the state-of-the-art solutions for accurately predicting the residential electricity load in the short term are mostly related to deep learning techniques with preprocessed augmented data: RNN, CNN, and transformers [138, 139, 140, 141, 135].

6.4.3 Market price forecasting

The last forecasting block to be discussed concerns the future day-ahead electricity market prices. Contrarily to the forecasting of power production and consumption, the market price predictions are not critical to the success of a dynamic pricing policy from the perspective of the supply side. Still, having access to quality forecasts for the future day-ahead market prices remains important in order to satisfy the constraints related to the profitability of the producer/retailer as well as the reduced electricity costs for the consumer. Once again, the predictions are assumed to be made just before the closing of the day-ahead market. Moreover, the uncertainty associated with these forecasts has to be taken into consideration.

Formally, the forecasting model related to the future day-ahead electricity market prices is denoted $\mathcal{F}_M$. Its input space $\mathcal{X}_M$ includes every single piece of information which may potentially explain the future electricity price on the day-ahead market for a certain hour. Its output space $\mathcal{Y}_M$ comprises a forecast of the day-ahead market price for that same hour of interest. Mathematically, both forecasting model input $x_t^M \in \mathcal{X}_M$ and output $y_t^M \in \mathcal{Y}_M$ at time step t can be expressed as follows:

$$x_t^M = \{x_t^P,\ x_t^C,\ G_t^F,\ M_t,\ I_t^M\}, \tag{6.14}$$

$$y_t^M = \lambda_t^F, \tag{6.15}$$

where:

- G_t^F represents forecasts about the state of the power grid as a whole (available production capacity, transmission lines, etc.) at the time step t,
- M_t represents diverse information in various markets related to energy (power, carbon, oil, gas, coal) in neighbouring geographical areas at time step t,
- I_t^M represents any extra piece of information that may potentially help to predict the future electricity price on the day-ahead market at time step t.

Once again, the scientific literature reveals that the state-of-the-art approach for day-ahead power market price forecasting mostly involves recent innovative machine learning techniques [142, 143, 144, 145, 146].

6.4.4 Demand response modelling

When it comes to designing a dynamic pricing policy from the perspective of the supply side, another essential algorithmic component is the mathematical modelling of the residential demand response to dynamic prices. Indeed, to make relevant dynamic pricing decisions, an estimation of the impact of the electricity price on the consumer's behaviour is necessary. More precisely, two important characteristics have to be studied:

The residential power consumption elasticity. This particular quantity measures the average percentage change of the residential power consumption in response to a percentage change in the electricity price. In other words, the elasticity captures the willingness of the consumer to adapt its behaviour when the price of electricity either increases or decreases. This elasticity is critical to the dynamic pricing approach, since it assesses the receptiveness of the consumers to dynamic prices. In fact, the residential power consumption elasticity can be considered as a quantitative indicator of the potential of the dynamic pricing approach.

The electricity load temporal dependence. Time plays an important role in power consumption. Firstly, the consumer's behaviour is highly dependent on the time of the day. The tendency to adapt this behaviour is also expected to be time-dependent. Therefore, the residential power consumption elasticity has to be a function of the time within a day, among other things. Secondly, a higher electricity price does not simply reduce the demand as with other commodities, but rather shifts part of the consumption earlier and/or later in time. This phenomenon reflects a complex temporal dependence for power consumption, which has to be accurately modelled in order to design a performing dynamic pricing policy.

Formally, the mathematical model of the residential demand response to dynamic prices is denoted $\mathcal{M}$. Its input space $\mathcal{X}_D$ is composed of the predicted power consumption before any demand response, of the dynamic prices to be sent to the consumers for several hours before and after the time period analysed, together with information about that time period. Its output space $\mathcal{Y}_D$ comprises the predicted power consumption after demand response to dynamic prices for that same time period. Mathematically, both demand response model input $x_t^D \in \mathcal{X}_D$ and output $y_t^D \in \mathcal{Y}_D$ at time step t can be expressed as the following:

$$x_t^D = \{C_t^F,\ Y_t',\ T_t\}, \tag{6.16}$$

$$y_t^D = c_t', \tag{6.17}$$

where $Y_t' = \{y_{t+\epsilon} \in \mathbb{R} \mid \epsilon = -k, ..., k\}$ is the dynamic price signal within a time window of size k and centred around time step t for which the demand response is analysed.

As far as the scientific literature about the modelling of demand response to dynamic prices is concerned, this interesting topic has not received a lot of attention from the research community yet. Still, there exist a few sound works presenting demand response models and assessing the receptiveness of the consumers to dynamic power prices [128, 129, 130, 131, 132], as explained in Section 6.2.

6.4.5 Uncertainty discussion

As previously hinted, a dynamic pricing policy has to make its decision on the basis of imperfect information. Indeed, multiple forecasts for the electricity price, production and consumption have to be generated 12 up to 35 hours in advance. Naturally, these predictions comes with a certain level of uncertainty that should not be neglected. Moreover, accurately modelling the residential demand response to dynamic prices is a particularly challenging task. Because of both the random human nature and the difficulty to perfectly capture the consumers' behaviour within a mathematical model, a notable level of uncertainty should also be taken into consideration at this stage. Therefore, multiple sources of uncertainty can be identified in the scope of the dynamic pricing decision-making problem at hand, and a proper management of this uncertainty is necessary.

A stochastic reasoning is recommended to make sound dynamic pricing decisions despite this substantial level of uncertainty. Typically, instead of considering each uncertain variable (production, consumption, market price, demand response) with a probability of 1, the full probability distributions behind these quantities have to be estimated and exploited. Based on this more complete information, the risk associated with uncertainty may be mitigated. Among others, such a claim refers to the methodology introduced in Chapter 5 of the thesis. Additionally, safety margins may also contribute to reduce this risk, but potentially at the expense of a lowered performance. In fact, there generally exists a trade-off between raw performance and risk, in line with the adage: *With great risk comes great reward.*

6.5 Performance assessment methodology

The present section introduces a methodology for quantitatively assessing the performance of a dynamic pricing policy in a comprehensive manner. As explained in Section 6.3.7, three disjoint objectives can be clearly identified. For the sake of completeness, this research work presents three quantitative indicators, one for each objective. The relative importance of these indicators is left to the discretion of the reader according to his or her main intention among the different goals previously defined.

Basically, the performance indicators introduced are based on the comparison with the original situation without any dynamic pricing mechanism. In this case, the power consumer is assumed to be fully ignorant about the mismatch problem between supply and demand. No information is provided to the customers of the producer/retailer, which consequently have an uninfluenced consumption behaviour. The price of electricity $\overline{e_t}$ is freely determined by the power producer/retailer. For instance, it could be a fixed tariff, or a price indexed on the day-ahead market price, in line with the following equation:

$$\overline{e_t} = \alpha \ \lambda_t + \beta \ , \tag{6.18}$$

where α and β are parameters to be set by the retailer of electricity.

Firstly, the impact of a dynamic pricing policy on the synchronisation between power supply and demand can be assessed through the performance indicator S quantifying the relative evolution of the deviation Δ_T. This quantity, expressed in %, can be mathematically expressed as the following:

$$S = 100 \ \frac{\overline{\Delta_T} - \Delta_T}{\overline{\Delta_T}} \ , \tag{6.19}$$

$$\overline{\Delta_T} = \sum_{t=0}^{T-1} |p_t - \overline{c_t}| \ , \tag{6.20}$$

where $\overline{\Delta_T}$ represents the lack of synchronisation between power supply and demand without any dynamic pricing mechanism. Therefore, the quantity S has to ideally be maximised, with a perfect synchronisation between supply and demand leading to a value of 100% reduction in deviation. On the contrary, a negative value for the indicator S reflects a deterioration of the situation, caused by the dynamic pricing policy.

Secondly, in a similar way, the consequences for the consumer regarding its electricity bill can be evaluated with the performance indicator B that informs about the relative evolution of this power bill. Also expressed in %, it can be mathematically computed as follows:

$$B = 100 \ \frac{\overline{B_T} - B_T}{\overline{B_T}} \ , \tag{6.21}$$

where $\overline{B_T} = \sum_{t=0}^{T-1} \overline{c_t} \ \overline{e_t}$ represents the electricity bill paid by the consumer in the absence of a dynamic pricing mechanism. Since the performance indicator B represents the percentage reduction in costs, it has to ideally be maximised as well.

Lastly, following the same methodology, the enhancement in terms of revenue for the producer/retailer can be efficiently quantified thanks to the performance indicator R. This quantity, expressed in %, represents the relative evolution of the producer/retailer's revenue and can be mathematically expressed as follows:

$$R = 100 \ \frac{R_T - \overline{R_T}}{\overline{R_T}} \ , \tag{6.22}$$

$$\overline{R_T} = \sum_{t=0}^{T-1} \left[\overline{c_t} \ \overline{e_t} - (c_t^F - p_t^F) \ \lambda_t - (\overline{c_t} - p_t) \ i_t \right] , \tag{6.23}$$

where $\overline{R_T}$ represents the producer/retailer's revenue without dynamic pricing. Following the same trend as the two previous performance indicators, the quantity R has to be maximised.

6.6 Conclusion

This thesis chapter presents a detailed formalisation of the decision-making problem faced by a producer/retailer willing to adopt a dynamic pricing approach, in order to induce an appropriate residential demand response. This is an important research work for the energy transition because it contributes to the development of intermittent renewable energy sources, by improving the synchronisation between their production and the power demand. Three key challenges are highlighted by this formalisation work to eventually achieve a practical solution. Firstly, the objective criterion to be maximised by a dynamic pricing policy is not trivially defined, since different goals that are not really compatible can be clearly identified. Secondly, the implementation of a performing dynamic pricing policy requires challenging algorithmic components: different forecasting blocks but also a mathematical model of the residential demand response to dynamic prices. Thirdly, the dynamic pricing decisions have to be made on the basis of imperfect information, because this particular decision-making problem is highly conditioned by the actual uncertainty about the future.

To end this thesis chapter, some avenues for future work are briefly discussed. In fact, the natural extension of the present research is to design, implement and evaluate in practice innovative dynamic pricing policies based on the formalisation carried out. This research work exclusively focuses on the philosophy and conceptual analysis of the approach, which is an essential step. Nevertheless, there remain diverse practical concerns that should be properly addressed to achieve a successful dynamic pricing decision-making policy. To go further, an exhaustive analysis of the scientific literature about each algorithmic component discussed in Section 6.4 is firstly welcomed, in order to identify and intelligently reproduce the state-of-the-art techniques within the context of interest. Secondly, different methodologies have to be thoroughly investigated for the design of the dynamic pricing policy itself. One can for instance mention, among others, stochastic optimisation techniques but also the deep reinforcement learning (RL) approach. In particular, a risk-sensitive distributional RL solution, as described in Chapter 5 could be a relevant fit. Finally, the dynamic pricing policies designed should be rigorously evaluated, analysed and compared by taking advantage of real-life experiments.

Education is the most powerful weapon which you can use to change the world.
— Nelson Mandela

Chapter 7

Conclusion

Figure 7.1: Illustration of the conclusion chapter of the thesis, created by a generative art AI [1].

7.1 Key contributions

To summarise, the ultimate objective of the present doctoral thesis is to delve into complex sequential decision-making problems related to markets and to design innovative algorithmic solutions by taking advantage of advanced AI techniques. In order to accomplish this goal, a comprehensive approach has been adopted, encompassing three distinct classes of research. Firstly, applied research is conducted, with a primary focus on developing novel AI-driven algorithmic solutions to tackle sequential decision-making problems encountered in energy and stock markets. Secondly, fundamental research is undertaken in distributional RL, which exhibits promising potential for effectively addressing the dynamics of market environments. Lastly, sustainable research is carried out, by formalising a decision-making problem within electricity markets, which is essential to drive the energy transition forward. In the following, the main contributions of the thesis are summarised, with additional details being available at the conclusion of each chapter.

The conducted applied research concentrates on three intricate sequential decision-making problems derived from the stock and the energy markets. One primary contribution of the thesis is the meticulous formalisation of each problem, accompanied by a comprehensive discussion of the key assumptions involved. Another major contribution lies in the design and elucidation of innovative AI-based algorithmic solutions aimed at effectively tackling the studied decision-making problems. Firstly, the long-term power procurement problem in the forward electricity markets is addressed, using a novel procurement policy relying on DL forecasting techniques and on the new mathematical concept of procurement uniformity. Secondly, an advanced DRL approach is presented as a promising solution to the algorithmic trading problem of determining the optimal trading position in the stock market. Lastly, the sequential decision-making problem associated with piloting a storage device in the intraday energy market is also tackled with a DRL methodology. To finish with the key contributions resulting from the applied research, the experiments and analyses conducted highlight the main limitations of the DRL approach in the context of market environments.

The fundamental research undertaken is motivated by the recognition of the limitations inherent in the DRL methodology when it comes to addressing decision-making problems within market environments. More precisely, the thesis makes three primary contributions to the distributional RL approach. Firstly, the research work introduces a novel distributional RL algorithm that leverages a particular DL architecture, which has been demonstrated to be a universal approximator of continuous monotonic functions. Moreover, this algorithm presents the significant advantage of supporting the learning of three, valid, and continuous representations of the random return distribution. Secondly, the doctoral thesis highlights an important limitation of a probability metric (Wasserstein distance) that is employed in several state-of-the-art distributional RL algorithms. Lastly, a novel intuitive methodology is presented for effectively learning risk-sensitive decision-making policies. This approach is specifically designed to be compatible with any distributional RL algorithm, and to enhance the interpretability of the resulting policies.

The last key contribution of the thesis is associated to the sustainable research carried out. This contribution involves the comprehensive formalisation of an important decision-making problem within the electricity markets in the context of the energy transition. In addition, a thorough discussion is provided regarding the essential algorithmic components required to effectively address this challenging problem. More precisely, the thesis investigates a dynamic pricing approach aimed at inducing an appropriate residential demand response, which could significantly improve the synchronisation between power consumption and the production of electricity from intermittent renewable energy sources.

7.2 Future work

This doctoral thesis represents a valuable contribution to the ongoing important research focused on addressing intricate real-world challenges using innovative DRL techniques. Even though the presented findings demonstrate promising advancements, there remains ample opportunity for further improvements. This section offers a concise summary of key concepts that could be considered and explored as avenues for future work. Further ideas and details may be found at the conclusion of each chapter of the dissertation.

First and foremost, a valuable undertaking would involve completing the research loop initiated by the thesis. This manuscript provides a comprehensive exploration of complex sequential decision-making problems related to markets. On the basis of the experiments conducted together with the resulting analyses, the thesis identifies key challenges associated with the application of the DRL approach in market environments. These important findings motivate the introduction of novel techniques based on the distributional RL methodology for addressing some of these challenges. In order to complete the research cycle, it would be interesting to rigorously assess the benefits of these new algorithmic solutions through their evaluation on the challenging market-related decision-making problems studied. The reason for this omission is the author's decision to allocate the remaining time to give the thesis a more sustainable dimension.

Secondly, future research endeavours are strongly encouraged to combat the extremely limited observability of market environments by augmenting the information available in the input space. Indeed, the poor observability of these environments significantly harms the performance of DRL algorithmic solutions. By allocating additional resources, a broader range of data could be investigated. In addition to incorporating technical and macroeconomic data, leveraging the information embedded within news holds substantial promise. This assertion arises from the recognition that markets ultimately reflect the expectations of market participants, which are generally influenced by or encapsulated within diverse news sources. In that regard, a well-designed AI-driven sentiment analysis solution is anticipated to offer valuable insights, by autonomously processing large volumes of news data.

Thirdly, risk-sensitive RL is an interesting area of research that deserves greater attention from the academic community. As elucidated in this thesis, effective risk management and mitigation are imperative across various fields of application, and particularly in markets. Consequently, there is a pressing demand for the development of more sophisticated risk-sensitive RL algorithms, alongside the establishment of insightful benchmarks to evaluate their performance. Furthermore, an augmentation of the theoretical foundations underlying risk-sensitive RL would undoubtedly contribute to a better comprehension and utilisation of these important techniques.

Fourthly, extensive research is required to alleviate a key weakness of the DRL approach: the black-box model. While DRL techniques excel at learning performing decision-making policies, they suffer from a severe lack of interpretability. In numerous fields of application including market contexts, this deficiency is not acceptable since it is capital to comprehend the decisions from a human standpoint and assess their reliability. Even though this thesis strives to contribute positively, substantial efforts are still required to solve this problem.

Fifthly, it is essential to enhance the adaptability of DRL techniques to establish them as a reliable solution for decision-making problems in market contexts. Indeed, the dynamic nature of market environments, which are continuously evolving and can even undergo major regime shifts, demands a high level of robustness. Therefore, further research is imperative to improve the capability of DRL models to learn and adapt in real-time, effectively updating the decision-making policies as new data progressively becomes available.

Lastly, the algorithmic solutions presented in this thesis are expected to be conveniently improved by integrating a multitude of recent advancements in AI and, more specifically, ML. A notable boost in performance is anticipated with more advanced DL models, such as residual connections, CNN, LSTM and transformers, among others. Moreover, the research community in RL has introduced various tricks and techniques to improve the original DRL algorithms. A great example is the Rainbow algorithm, which is a compilation of diverse enhancements to the DQN algorithm. By adopting a similar approach and considering these various augmentations, a substantial increase in performance may potentially be achieved for the algorithmic solutions introduced.

To end this section, it is important to acknowledge that the aforementioned exploration of ideas for future works is not an exhaustive compilation. The research topic examined, which involves the application of the DRL approach to real-world issues, is particularly extensive in its scope. Consequently, there is a multitude of other opportunities for contributing to this important research endeavour.

7.3 Author's closing words

In concluding the present doctoral thesis, I am filled with a profound sense of satisfaction. The arduous journey I undertook, the countless hours of research, and the myriad challenges that I encountered along the way have ultimately shaped me into the person I am today. This extraordinarily fulfilling experience has been nothing short of transformative, leaving an enduring imprint on my intellectual and personal growth. In addition to greatly expanding my knowledge and expertise, pursuing a doctoral degree has significantly fostered resilience, perseverance, and critical thinking, among others. Without a doubt, this background will prove invaluable in shaping my future endeavours.

This thesis stands as the culmination of years of dedicated research. Besides the numerous personal benefits, it is my sincere hope that the insights presented within these pages will serve as a catalyst for further research, innovation, and transformative action. I strongly believe that the sharing of knowledge possesses the power to bring about substantial and meaningful change. May this research work inspire fellow researchers, ignite new avenues of inquiry, and pave the way for breakthroughs yet to come. At the very least, I am profoundly thankful to the readers for their interest in my research work.

As I bid farewell to this influential chapter of my professional life, I eagerly embrace the prospects that lie ahead. Among the multitude of opportunities, I am particularly motivated to contribute to one of the paramount challenges confronting humanity in the 21st century: the energy transition. This aligns perfectly with my unwavering convictions to combat climate change and strive for a more sustainable and resilient future...

Bibliography

[1] Midjourney. `https://midjourney.com`, 2023. Accessed: 2023-02-07.

[2] Thibaut Théate, Sébastien Mathieu, and Damien Ernst. An artificial intelligence solution for electricity procurement in forward markets. *Energies*, 13(23), 2020.

[3] Thibaut Théate and Damien Ernst. An application of deep reinforcement learning to algorithmic trading. *Expert Systems with Applications*, 173:114632, 2021.

[4] Thibaut Théate, Antoine Wehenkel, Adrien Bolland, Gilles Louppe, and Damien Ernst. Distributional reinforcement learning with unconstrained monotonic neural networks. *Neurocomputing*, 534:199–219, 2023.

[5] Thibaut Théate and Damien Ernst. Risk-sensitive policy with distributional reinforcement learning. *Algorithms*, 16(7):325, 2023.

[6] Thibaut Théate, Antonio Sutera, and Damien Ernst. Matching of everyday power supply and demand with dynamic pricing: Problem formalisation and conceptual analysis. *Energy Reports*, 9:2453–2462, 2023.

[7] Ioannis Boukas, Damien Ernst, Thibaut Théate, Adrien Bolland, Alexandre Huynen, Martin Buchwald, Christelle Wynants, and Bertrand Cornélusse. A deep reinforcement learning framework for continuous intraday market bidding. *Machine Learning*, 110(9):2335–2387, 2021.

[8] Miguel Carrion, Andy B. Philpott, Antonio J. Conejo, and Jos M. Arroyo. A stochastic programming approach to electric energy procurement for large consumers. *IEEE Transactions on Power Systems*, 22(2):744–754, 2007.

[9] Antonio J. Conejo, Miguel Carrion, and Juan M. Morales. *Decision Making under Uncertainty in Electricity Markets*, volume 1. Springer US, 2010.

[10] Kazem Zare, Mohsen Parsa Moghaddam, and Mohammad Kazem Sheikh El Eslami. Electricity procurement for large consumers based on information gap decision theory. *Energy Policy*, 38(1):234–242, 2010.

[11] Sayyad Nojavan, Behnam Mohammadi-Ivatloo, and Kazem Zare. Robust optimization based price-taker retailer bidding strategy under pool market price uncertainty. *International Journal of Electrical Power & Energy Systems*, 73:955–963, 2015.

[12] Patrizia Beraldi, Antonio Violi, Maria Elena Bruni, and Gianluca Carrozzino. A probabilistically constrained approach for the energy procurement problem. *Energies*, 10(12):2179, 2017.

[13] Patrizia Beraldi, Antonio Violi, Gianluca Carrozzino, and Maria Elena Bruni. The optimal electric energy procurement problem under reliability constraints. *Energy Procedia*, 136:283–289, 2017.

[14] Chiyuan Zhang, Oriol Vinyals, Rémi Munos, and Samy Bengio. A Study on Overfitting in Deep Reinforcement Learning. *CoRR*, abs/1804.06893, 2018.

[15] Feihu Hu, X. Feng, and Hui Cao. A short-term decision model for electricity retailers: Electricity procurement and time-of-use pricing. *Energies*, 11:3258, 2018.

[16] R. Konishi, Akiko Takeda, and M. Takahashi. Optimal sizing of energy storage systems for the energy procurement problem in multi-period markets under uncertainties. *Energies*, 11:158, 2018.

[17] T. Wang and S. Deng. Multi-period energy procurement policies for smart-grid communities with deferrable demand and supplementary uncertain power supplies. *Omega-international Journal of Management Science*, 89:212–226, 2019.

[18] J. Zhang, Y. Zheng, M. Yao, Huiji Wang, and Z. Hu. An agent-based two-stage trading model for direct electricity procurement of large consumers. *Sustainability*, 11:5031, 2019.

[19] Ernest P. Chan. *Quantitative Trading: How to Build Your Own Algorithmic Trading Business*. Wiley, 2009.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016.

[21] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. *ICML*, 30, 2013.

[22] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[23] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.

[24] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980, 2015.

[25] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep Learning. *Nature*, 521, 2015.

[26] Ice Endex Belgian Power Base Futures. `https://www.theice.com/products/27993084/Belgian-Power-Base-Futures`, 2020. Accessed: 2020-05-12.

[27] Hassan Ismail Fawaz, G. Forestier, Jonathan Weber, L. Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33:917–963, 2019.

[28] Omer Berat Sezer, M. U. Gudelek, and A. Ozbayoglu. Financial time series forecasting with deep learning : A systematic literature review: 2005-2019. *ArXiv*, abs/1911.13288, 2020.

[29] Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. Object recognition with gradient-based learning. In *Shape, Contour and Grouping in Computer Vision*, volume 1681, page 319. Springer, 1999.

[30] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.

[32] Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5244–5254, 2019.

[33] Bryan Lim, Sercan Ömer Arik, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *CoRR*, abs/1912.09363, 2019.

[34] Yuxi Li. Deep Reinforcement Learning: An Overview. *CoRR*, abs/1701.07274, 2017.

[35] Ernest P. Chan. *Algorithmic Trading: Winning Strategies and Their Rationale*. Wiley, 2013.

[36] Rishi K. Narang. *Inside the Black Box*. Wiley, 2009.

[37] Andrés Arévalo, Jaime Niño, Germán Hernández, and Javier Sandoval. High-Frequency Trading Strategy Based on Deep Neural Networks. *ICIC*, 2016.

[38] Wei Ning Bao, Jun Yue, and Yulei Rao. A Deep Learning Framework for Financial Time Series using Stacked Autoencoders and Long-Short Term Memory. *PloS one*, 12, 2017.

[39] John E. Moody and Matthew Saffell. Learning to Trade via Direct Reinforcement. *IEEE transactions on neural networks*, 12 4:875–89, 2001.

[40] Michael A. H. Dempster and V. Leemans. An Automated FX Trading System using Adaptive Reinforcement Learning. *Expert Syst. Appl.*, 30:543–552, 2006.

[41] Yue Deng, Feng Bao, Youyong Kong, Zhiquan Ren, and Qionghai Dai. Deep Direct Reinforcement Learning for Financial Signal Representation and Trading. *IEEE Transactions on Neural Networks and Learning Systems*, 28:653–664, 2017.

[42] João Carapuço, Rui Ferreira Neves, and Nuno Horta. Reinforcement Learning applied to Forex Trading. *Appl. Soft Comput.*, 73:783–794, 2018.

[43] John P. A. Ioannidis. Why Most Published Research Findings Are False. *PLoS Med*, 2:124, 2005.

[44] David H. Bailey, Jonathan M. Borwein, Marcos López de Prado, and Qiji Jim Zhu. Pseudo-Mathematics and Financial Charlatanism: The Effects of Backtest Overfitting on Out-of-Sample Performance. *Notice of the American Mathematical Society*, pages 458–471, 2014.

[45] Terrence Hendershott, Charles Michael Jones, and Albert J. Menkveld. Does Algorithmic Trading Improve Liquidity? *Journal of Finance*, 66:1–33, 2011.

[46] Philip C. Treleaven, Michal Galas, and Vidhi Lalchand. Algorithmic Trading Review. *Commun. ACM*, 56:76–85, 2013.

[47] Giuseppe Nuti, Mahnoosh Mirghaemi, Philip C. Treleaven, and Chaiyakorn Yingsaeree. Algorithmic Trading. *Computer*, 44:61–69, 2011.

[48] David Leinweber and Jacob Sisk. Event-Driven Trading and the "New News". *The Journal of Portfolio Management*, 38:110–124, 2011.

[49] Johan Bollen, Huina Mao, and Xiao jun Zeng. Twitter Mood Predicts the Stock Market. *J. Comput. Science*, 2:1–8, 2011.

[50] Wijnand Nuij, Viorel Milea, Frederik Hogenboom, Flavius Frasincar, and Uzay Kaymak. An Automated Framework for Incorporating News into Stock Trading Strategies. *IEEE Transactions on Knowledge and Data Engineering*, 26:823–835, 2014.

[51] Christopher J. C. H. Watkins and Peter Dayan. Technical note: Q-learning. *Machine Learning*, 8:279–292, 1992.

[52] Richard Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[53] Marc G. Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: an evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

[54] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing Atari with Deep Reinforcement Learning. *CoRR*, abs/1312.5602, 2013.

[55] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[56] Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Gheshlaghi Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 3215–3222. AAAI Press, 2018.

[57] Richard S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44, 1988.

[58] Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double Q-learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2094–2100. AAAI Press, 2016.

[59] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *4th International Conference on Learning Representations, ICLR*, 2016.

[60] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado van Hasselt, Marc Lanctot, and Nando de Freitas. Dueling network architectures for deep reinforcement learning. In *Proceedings of the 33nd International Conference on Machine Learning, ICML*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1995–2003. JMLR.org, 2016.

[61] Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 449–458. PMLR, 2017.

[62] Meire Fortunato, Mohammad Gheshlaghi Azar, Bilal Piot, Jacob Menick, Matteo Hessel, Ian Osband, Alex Graves, Volodymyr Mnih, Rémi Munos, Demis Hassabis, Olivier Pietquin, Charles Blundell, and Shane Legg. Noisy networks for exploration. In *6th International Conference on Learning Representations, ICLR*. OpenReview.net, 2018.

[63] Ian J. Goodfellow, Yoshua Bengio, and Aaron C. Courville. Deep Learning. *Nature*, 521:436–444, 2015.

[64] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2018.

[65] Csaba Szepesvari. *Algorithms for Reinforcement Learning*. Morgan and Claypool Publishers, 2010.

[66] Lucian Busoniu, Robert Babuska, Bart De Schutter, and Damien Ernst. *Reinforcement Learning and Dynamic Programming using Function Approximators*. CRC Press, 2010.

[67] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. A Brief Survey of Deep Reinforcement Learning. *CoRR*, abs/1708.05866, 2017.

[68] Kun Shao, Zhentao Tang, Yuanheng Zhu, Nannan Li, and Dongbin Zhao. A Survey of Deep Reinforcement Learning in Video Games. *ArXiv*, abs/1912.10944, 2019.

[69] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *CoRR*, abs/1502.03167, 2015.

[70] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Vedavyas Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy P. Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the Game of Go with Deep Neural Networks and Tree Search. *Nature*, 529:484–489, 2016.

[71] Matthew J. Hausknecht and Peter Stone. Deep Recurrent Q-Learning for Partially Observable MDPs. *CoRR*, abs/1507.06527, 2015.

[72] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. *CoRR*, abs/1707.06347, 2017.

[73] Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning, ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 449–458. PMLR, 2017.

[74] Clare Lyle, Pablo Samuel Castro, and Marc G. Bellemare. A comparative analysis of expected and distributional reinforcement learning. In *AAAI*, 2019.

[75] Antoine Wehenkel and Gilles Louppe. Unconstrained monotonic neural networks. In *Advances in Neural Information Processing Systems 32, NeurIPS*, pages 1543–1553, 2019.

[76] Amir Rahimi, Amirreza Shaban, Ching-An Cheng, Richard Hartley, and Byron Boots. Intra order-preserving functions for calibration of multi-class neural networks. In *Advances in Neural Information Processing Systems 33, NeurIPS*, 2020.

[77] Yaoshu Wang, Chuan Xiao, Jianbin Qin, Rui Mao, Makoto Onizuka, Wei Wang, and Rui Zhang. Consistent and flexible selectivity estimation for high-dimensional data. *CoRR*, abs/2005.09908, 2020.

[78] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on Machine Learning, ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1530–1538. JMLR.org, 2015.

[79] Mark Rowland, Robert Dadashi, Saurabh Kumar, Rémi Munos, Marc G. Bellemare, and Will Dabney. Statistics and samples in distributional reinforcement learning. In *ICML*, 2019.

[80] Mark Rowland, Marc G. Bellemare, Will Dabney, Rémi Munos, and Yee Whye Teh. An analysis of categorical distributional reinforcement learning. In *International Conference on Artificial Intelligence and Statistics, AISTATS*, volume 84 of *Proceedings of Machine Learning Research*, pages 29–37. PMLR, 2018.

[81] Will Dabney, Mark Rowland, Marc G. Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, AAAI*, pages 2892–2901. AAAI Press, 2018.

[82] Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. Implicit quantile networks for distributional reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning, ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 1104–1113. PMLR, 2018.

[83] Derek Yang, Li Zhao, Zichuan Lin, Tao Qin, Jiang Bian, and Tie-Yan Liu. Fully parameterized quantile function for distributional reinforcement learning. In *Advances in Neural Information Processing Systems 32, NeurIPS*, pages 6190–6199, 2019.

[84] Thanh Nguyen-Tang, Sunil Gupta, and Svetha Venkatesh. Distributional reinforcement learning via moment matching. In *AAAI*, 2021.

[85] Fan Zhou, Jianing Wang, and Xingdong Feng. Non-crossing quantile regression for distributional reinforcement learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[86] Will Dabney, Zeb Kurth-Nelson, Naoshige Uchida, Clara Kwon Starkweather, Demis Hassabis, Rémi Munos, and Matthew M. Botvinick. A distributional code for value in dopamine-based reinforcement learning. *Nature*, 577:671–675, 2020.

[87] Tetsuro Morimura, Masashi Sugiyama, Hisashi Kashima, Hirotaka Hachiya, and Toshiyuki Tanaka. Parametric return density estimation for reinforcement learning. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence, UAI*, pages 368–375. AUAI Press, 2010.

[88] Roger Koenker. *Quantile Regression*. Cambridge University Press, 2005.

[89] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. *CoRR*, abs/1606.01540, 2016.

[90] Johan Samir Obando-Ceron and Pablo Samuel Castro. Revisiting rainbow: Promoting more insightful and inclusive deep reinforcement learning research. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 1373–1383. PMLR, 2021.

[91] Marlos C. Machado, Marc G. Bellemare, Erik Talvitie, Joel Veness, Matthew J. Hausknecht, and Michael Bowling. Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents. *Journal of Artificial Intelligence Research*, 61:523–562, 2018.

[92] Kenny Young and Tian Tian. MinAtar: An Atari-inspired testbed for thorough and reproducible reinforcement learning experiments. *CoRR*, abs/1903.03176, 2019.

[93] Gabriel Dulac-Arnold, Nir Levine, Daniel J. Mankowitz, Jerry Li, Cosmin Paduraru, Sven Gowal, and Todd Hester. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Machine Learning*, 110(9):2419–2468, 2021.

[94] Omer Gottesman, Fredrik D. Johansson, Matthieu Komorowski, Aldo A. Faisal, David Sontag, Finale Doshi-Velez, and Leo Anthony Celi. Guidelines for reinforcement learning in healthcare. *Nature Medicine*, 25:16–18, 2019.

[95] Brijen Thananjeyan, Ashwin Balakrishna, Suraj Nair, Michael Luo, Krishnan Srinivasan, Minho Hwang, Joseph E. Gonzalez, Julian Ibarz, Chelsea Finn, and Ken Goldberg. Recovery RL: Safe reinforcement learning with learned recovery zones. *IEEE Robotics and Automation Letters*, 6(3):4915–4922, 2021.

[96] Zeyu Zhu and Huijing Zhao. A survey of deep RL and IL for autonomous driving policy learning. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):14043–14065, 2022.

[97] Javier García and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16:1437–1480, 2015.

[98] Dotan Di Castro, Aviv Tamar, and Shie Mannor. Policy gradients with variance related risk criteria. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress, 2012.

[99] Prashanth L. A. and Mohammad Ghavamzadeh. Actor-Critic algorithms for risk-sensitive MDPs. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 252–260, 2013.

[100] Shangtong Zhang, Bo Liu, and Shimon Whiteson. Mean-variance policy iteration for risk-averse reinforcement learning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 10905–10913. AAAI Press, 2021.

[101] R. Tyrrell Rockafellar and Stanislav Uryasev. Conditional value-at-risk for general loss distributions. *Corporate Finance and Organizations eJournal*, 2001.

[102] Yinlam Chow, Aviv Tamar, Shie Mannor, and Marco Pavone. Risk-sensitive and robust decision-making: a CVaR optimization approach. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1522–1530, 2015.

[103] Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. Risk-constrained reinforcement learning with percentile risk criteria. *Journal of Machine Learning Research*, 18:167:1–167:51, 2017.

[104] Aviv Tamar, Yonatan Glassner, and Shie Mannor. Optimizing the CVaR via sampling. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pages 2993–2999. AAAI Press, 2015.

[105] Aravind Rajeswaran, Sarvjeet Ghotra, Balaraman Ravindran, and Sergey Levine. EPOpt: Learning robust neural network policies using model ensembles. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[106] Takuya Hiraoka, Takahisa Imagawa, Tatsuya Mori, Takashi Onishi, and Yoshimasa Tsuruoka. Learning robust options by conditional value at risk optimization. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 2615–2625, 2019.

[107] Yun Shen, Michael J. Tobia, Tobias Sommer, and Klaus Obermayer. Risk-sensitive reinforcement learning. *Neural Computation*, 26(7):1298–1328, 2014.

[108] Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. Implicit quantile networks for distributional reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1104–1113. PMLR, 2018.

[109] Yichuan Charlie Tang, Jian Zhang, and Ruslan Salakhutdinov. Worst cases policy gradients. In *3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings*, volume 100 of *Proceedings of Machine Learning Research*, pages 1078–1093. PMLR, 2019.

[110] Núria Armengol Urpí, Sebastian Curi, and Andreas Krause. Risk-averse offline reinforcement learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[111] Qisong Yang, T. D. Simão, Simon Tindemans, and Matthijs T. J. Spaan. Safety-constrained reinforcement learning with a distributional safety critic. *Machine Learning*, 2022.

[112] Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 2817–2826. PMLR, 2017.

[113] Wei Qiu, Xinrun Wang, Runsheng Yu, Rundong Wang, Xu He, Bo An, Svetlana Obraztsova, and Zinovi Rabinovich. RMIX: learning risk-sensitive policies for cooperative reinforcement learning agents. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 23049–23062, 2021.

[114] IPCC. *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 2021.

[115] Hannah Ritchie and Max Roser. Energy. *Our World in Data*, 2020. , https://ourworldindata.org/energy.

[116] Spyros Chatzivasileiadis, Damien Ernst, and Göran Andersson. The global grid. *Renewable Energy*, 57:372–383, 2013.

[117] Noah Kittner, Felix Lill, and Daniel M. Kammen. Energy storage deployment and innovation for the clean energy transition. *Nature Energy*, 2:17125, 2017.

[118] Peter Palensky and Dietmar Dietrich. Demand side management: Demand response, intelligent energy systems, and smart loads. *IEEE Transactions on Industrial Informatics*, 7:381–388, 2011.

[119] Pierluigi Siano. Demand response and smart grids — A survey. *Renewable & Sustainable Energy Reviews*, 30:461–478, 2014.

[120] Ruilong Deng, Zaiyue Yang, Mo-Yuen Chow, and Jiming Chen. A survey on demand response in smart grids: Mathematical models and approaches. *IEEE Transactions on Industrial Informatics*, 11:570–582, 2015.

[121] John S. Vardakas, Nizar Zorba, and Christos V. Verikoukis. A survey on demand response programs in smart grids: Pricing methods and optimization algorithms. *IEEE Communications Surveys & Tutorials*, 17:152–178, 2015.

[122] Haider Tarish Haider, Ong Hang See, and Wilfried Elmenreich. A review of residential demand response of smart grid. *Renewable & Sustainable Energy Reviews*, 59:166–178, 2016.

[123] Zhuang Zhao, Won Cheol Lee, Yoan Shin, and Kyung-Bin Song. An optimal power scheduling method for demand response in home energy management system. *IEEE Transactions on Smart Grid*, 4:1391–1400, 2013.

[124] Matteo Muratori and Giorgio Rizzoni. Residential demand response: Dynamic energy management and time-varying electricity pricing. *IEEE Transactions on Power Systems*, 31:1108–1117, 2016.

[125] Nian Liu, Xinghuo Yu, Cheng Wang, Chaojie Li, Li Ma, and Jinyong Lei. Energy-sharing model with price-based demand response for microgrids of peer-to-peer prosumers. *IEEE Transactions on Power Systems*, 32:3569–3583, 2017.

[126] José R. Vázquez-Canteli and Zoltán Nagy. Reinforcement learning for demand response: A review of algorithms and modeling techniques. *Applied Energy*, 2019.

[127] Hepeng Li, Zhiqiang Wan, and Haibo He. Real-time residential demand response. *IEEE Transactions on Smart Grid*, 11:4144–4154, 2020.

[128] Sebastian Gottwalt, Wolfgang Ketter, Carsten Block, John Collins, and Christof Weinhardt. Demand side management — A simulation of household behavior under variable prices. *Energy Policy*, 39:8163–8174, 2011.

[129] Margaux Brégère and Ricardo J. Bessa. Simulating tariff impact in electrical energy consumption profiles with conditional variational autoencoders. *IEEE Access*, 8:131949–131966, 2020.

[130] Kamalanathan Ganesan, João Tomé Saraiva, and Ricardo J. Bessa. Functional model of residential consumption elasticity under dynamic tariffs. *Energy and Buildings*, 255, 2022.

[131] Yongxiu He, Bing Wang, Jianhui Wang, Jianhui Wang, Wei Xiong, and Tian Xia. Residential demand response behavior analysis based on Monte Carlo simulation: The case of Yinchuan in China. *Energy*, 47:230–236, 2012.

[132] E. A. M. Klaassen, C. B. A. Kobus, Jasper Frunt, and Johannes G. Slootweg. Responsiveness of residential electricity demand to dynamic tariffs: Experiences from a large field test in the Netherlands. *Applied Energy*, 183:1065–1074, 2016.

[133] Conor Sweeney, Ricardo J. Bessa, Jethro Browell, and Pierre Pinson. The future of forecasting for renewable energy. *WIREs Energy and Environment*, 2019.

[134] Rizwan Ahmed, Victor Sreeram, Yogendra D. Mishra, and Muhammad Arif. A review and evaluation of the state-of-the-art in PV solar power forecasting: Techniques and optimization. *Renewable & Sustainable Energy Reviews*, 124:109792, 2020.

[135] Sheraz Aslam, Herodotos Herodotou, Syed Muhammad Mohsin, Nadeem Javaid, Nouman Ashraf, and Shahzad Aslam. A survey on deep learning methods for power load and renewable energy forecasting in smart microgrids. *Renewable and Sustainable Energy Reviews*, 144, 2021.

[136] Hamidreza Jahangir, Hanif Tayarani, Saleh Sadeghi Gougheri, Masoud Aliakbar Golkar, Ali Ahmadian, and Ali Elkamel. Deep learning-based forecasting approach in smart grids with microclustering and bidirectional LSTM network. *IEEE Transactions on Industrial Electronics*, 68:8298–8309, 2021.

[137] Detlev Heinemann, Elke Lorenz, and Marco Girodo. Forecasting solar radiation. *Journal of Cases on Information Technology*, 2021.

[138] Weicong Kong, Zhao Yang Dong, Youwei Jia, David. J. Hill, Yan Xu, and Yuan Zhang. Short-term residential load forecasting based on LSTM recurrent neural network. *IEEE Transactions on Smart Grid*, 10:841–851, 2019.

[139] Nivethitha Somu, Gauthama Raman M R, and Krithi Ramamritham. A hybrid model for building energy consumption forecasting using long short term memory networks. *Applied Energy*, 261:114131, 2020.

[140] Xue bo Jin, Weiguang Zheng, Jianlei Kong, Xiaoyi Wang, Yuting Bai, Tingli Su, and Seng Lin. Deep-learning forecasting method for electric power load via attention-based encoder-decoder with bayesian optimization. *Energies*, 14:1596, 2021.

[141] Alberto Gasparin, Slobodan Lukovic, and Cesare Alippi. Deep learning for time series forecasting: The electric load case. *CAAI Transactions on Intelligence Technology*, 7, 2021.

[142] Rafał Weron. Electricity price forecasting: A review of the state-of-the-art with a look into the future. *HSC Research Reports*, 2014.

[143] Jakub Nowotarski and Rafał Weron. Recent advances in electricity price forecasting: A review of probabilistic forecasting. *HSC Research Reports*, 2016.

[144] Abbas Rahimi Gollou and Noradin Ghadimi. A new feature selection and hybrid forecast engine for day-ahead price forecasting of electricity markets. *J. Intell. Fuzzy Syst.*, 32:4031–4045, 2017.

[145] Umut Ugurlu, Ilkay Oksuz, and Oktay Tas. Electricity price forecasting using recurrent neural networks. *Energies*, 11:1255, 2018.

[146] Hamidreza Jahangir, Hanif Tayarani, Sina Baghali, Ali Ahmadian, Ali Elkamel, Masoud Aliakbar Golkar, and Miguel Castilla. A novel electricity price forecasting approach based on dimension reduction strategy and rough artificial neural networks. *IEEE Transactions on Industrial Informatics*, 16:2369–2381, 2020.