DEGREE PROJECT IN TECHNOLOGY,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2023

Reinforcement learning for EV charging optimization

A holistic perspective for commercial vehicle fleets

Enzo Alexander Cording

TRITA-ITM-EX 2023:524

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF INDUSTRIAL ENGINEERING AND MANAGEMENT

Author

Enzo Alexander Cording - cording@kth.se
School of Industrial Engineering and Management
KTH Royal Institute of Technology
TRITA-ITM-EX 2023:524

Examiner

Viktoria Martin
Stockholm
KTH Royal Institute of Technology

Supervisor

Jagruti Ramsing Thakur

Stockholm

KTH Royal Institute of Technology

Abstract

Recent years have seen an unprecedented uptake in electric vehicles, driven by the global push to reduce carbon emissions. At the same time, intermittent renewables are being deployed increasingly. These developments are putting flexibility measures such as dynamic load management in the spotlight of the energy transition. Flexibility measures must consider EV charging, as it has the ability to introduce grid constraints: In Germany, the cumulative power of all EV onboard chargers amounts to ca. 120 GW, while the German peak load only amounts to 80 GW. Commercial operations have strong incentives to optimize charging and flatten peak loads in real-time, given that the highest quarter-hour can determine the power-related energy bill, and that a blown fuse due to overloading can halt operations. Increasing research efforts have therefore gone into real-time-capable optimization methods. Reinforcement Learning (RL) has particularly gained attention due to its versatility, performance and real-time capabilities. This thesis implements such an approach and introduces FleetRL as a realistic RL environment for EV charging, with a focus on commercial vehicle fleets. Through its implementation, it was found that RL saved up to 83% compared to static benchmarks, and that grid overloading was entirely avoided in some scenarios by sacrificing small portions of SOC, or by delaying the charging process. Linear optimization with one year of perfect knowledge outperformed RL, but reached its practical limits in one use-case, where a feasible solution could not be found by the solver. Overall, this thesis makes a strong case for RL-based EV charging. It further provides a foundation which can be built upon: a modular, open-source software framework that integrates an MDP model, schedule generation, and non-linear battery degradation.

Keywords

Deep Reinforcement Learning, EV charging optimization, Artificial Intelligence, Commercial vehicle fleets, Electric vehicles

Sammanfattning

Elektrifieringen av transportsektorn är en nödvändig men utmanande uppgift. I kombination med ökande solcellsproduktion och förnybara energikällor skapar det ett dilemma för elnätet som kräver omfattande flexibilitetsåtgärder. Dessa åtgärder måste inkludera laddning av elbilar, ett fenomen som har lett till aldrig tidigare skådade belastningstoppar. Ur ett kommersiellt perspektiv är incitamentet att optimera laddningsprocessen och säkerställa drifttid. Forskningen har fokuserat på realtidsoptimeringsmetoder som Deep Reinforcement Learning (DRL). Denna avhandling introducerar FleetRL som en ny RL-miljö för EV-laddning av kommersiella flottor. Genom att tillämpa ramverket visade det sig att RL sparade upp till 83% jämfört med statiska riktmärken, och att överbelastning av nätet helt kunde undvikas i de flesta scenarier. Linjär optimering överträffade RL men nådde sina gränser i snävt begränsade användningsfall. Efter att ha funnit ett positivt business case för varje kommersiellt användningsområde, ger denna avhandling ett starkt argument för RL-baserad laddning och en grund för framtida arbete via praktiska insikter och ett modulärt mjukvaruramverk med öppen källkod.

Nyckelord

Deep Reinforcement Learning, optimering av elbilsladdning, artificiell intelligens, kommersiella fordonsflottor, Elektriska fordon

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Jagruti Thakur, for her unwavering support and guidance throughout this research journey. Her expertise and valuable insights have been essential for my work, and her thoughtful supervision shaped my understanding of methodological research, EV charging optimization and its real-world applications.

I would like to thank my examiner, Viktoria Martin, whose critical reviews and feedback greatly contributed to the integrity and quality of this thesis.

Special thanks are also given to Fernando Gallego, Cristian Martin, and Manuel Diaz from the University of Malaga and the EVOLVE research project on applications of RL in EV charging. Their experience with RL, and our theoretical discussions, have been instrumental in broadening my domain knowledge.

Lucas Koltermann from ISEA at RWTH Aachen also deserves a particular mention. Being my Bachelor thesis supervisor, he took the time for discussions on the German EV market and its developments. His insights have been incredibly informative and provided a unique perspective to my research.

I also wish to acknowledge Xavier Weiss and Hussain Kazmi for their mentorship during the EIT Data Science course. Their insights and interesting discussions on RL, and AI's future have been inspiring, and provided a new point of view.

Last but not least, I would like to express sincere gratitude to my brother, Henry, who not only lent me his gaming PC at night to train RL models, but also shared his insights into object-orientated programming that provided me with invaluable guidance during my first extensive programming project.

Finally, I would like to thank my family and friends for their continuous support during this project and my endeavours.

Acronyms

RL Reinforcement Learning
DRL Deep Reinforcement Learning
DQL Deep Q-learning
DDPG Deep Deterministic Policy Gradient
DQN Deep Q-Network
DNN Deep Neural Network
CNN Convolutional Neural Network
ML Machine Learning
MDP Markov Decision Process
BRL FQI Batch-RL with fitted Q-iteration
TD3 Twin Delayed Deep Deterministic Policy Gradient
SAC Soft Actor Critic
PPO Proximal Policy Optimization
MARL Multi-Agent Reinforcement Learning
LSTM Long Short-Term Memory
DSO Distribution Service Operator
TD Temporal Difference
MC Monte Carlo
DP Dynamic Programming
SB3 Stable-Baselines3
MLP Multi Layer Perceptron
MSE Mean Squared Error
LP Linear Programming
MILP Mixed Integer Linear Programming
MPC Model-Predictive Control
EV Electric Vehicle

PHEV Plug-in Hybrid Electric Vehicle

EVSE EV supply equipment

SOC State of Charge

SOH State of Health

NPV Net Present Value

UC Uncontrolled Charging

Table of contents

Chapter 1

Introduction

Road transport makes up 17.8% of global emissions - its electrification is thus one of the most impactful measures to achieve greenhouse gas and pollution reduction in urban areas [1–4]. Electric vehicles (EVs) are being widely adopted in households, public transport, and companies. Besides their merits, however, EVs introduce challenges to the energy system as it is known today. EV charging introduces unpredictable load peaks; in combination with the rapidly growing EV adoption rate, this places an increasing strain on the electricity grid. This problem is also experienced on an end-user level: if not managed properly, the simultaneous charging of an electric vehicle and use of multiple household appliances can trip the house's breaker box. For commercial fleets it is even more important to manage the charging process, since parallel charging, vehicle down-time, and operational costs must be dealt with. These various challenges necessitate an optimization of the charging process for electric vehicles, especially in the commercial space.

EV charging optimization has been abundantly studied in scientific literature. Generally speaking, the objective is to charge the vehicles in the most cost- or operationally-effective manner whilst satisfying the constraints of the specific use-case. These constraints can for example include the grid connection, the driving schedules, and the vehicle range. Solving a constrained optimization is commonly addressed with linear programming techniques, such as mixed integer linear programming [5–7]. The problem of EV charging, however, can become complex: driving behaviour can be random, market prices change over time and non-linearities can be introduced to the scope of the optimization. Where linear

methods fall short, deterministic, non-linear mathematical models can describe complex relationships. However, they have drawbacks when it comes to real-time applications due to computing time. Given that the EV charging problem features random, instantly occurring elements such as random behaviour or line overloading, real-time capable algorithms hold considerable value. This led to research in the field of reinforcement learning (RL) which has been shown to be more suitable for real-time use [8]. Current contemporary research in this area has largely remained in the theoretical domain and a transition towards real-world applications is yet to be made. The potential benefits of RL and the lack of real-world applications motivate this thesis to propose a novel framework that incorporates RL and an economic analysis of the required charging, metering, and communication infrastructure [9].

A thorough literature review forms the basis of this study, focusing on commercial fleets, charging management systems and applications of RL in EV charging optimization in Chapter 2. From the identified research gaps, scope and objectives are formulated; this is done in Chapter 3. Subsequently, the most important theory of Reinforcement Learning (RL) is explained in Chapter 4, covering the fundamentals and the models used in this study. Chapter 5 covers the methodology: designing the use-cases, implementing RL-based optimization, and conducting an economic impact assessment. Chapter 6 presents and discuss the results, followed by a final conclusion in Chapter 7. Figure 1.0.1 summarizes the thesis structure graphically.

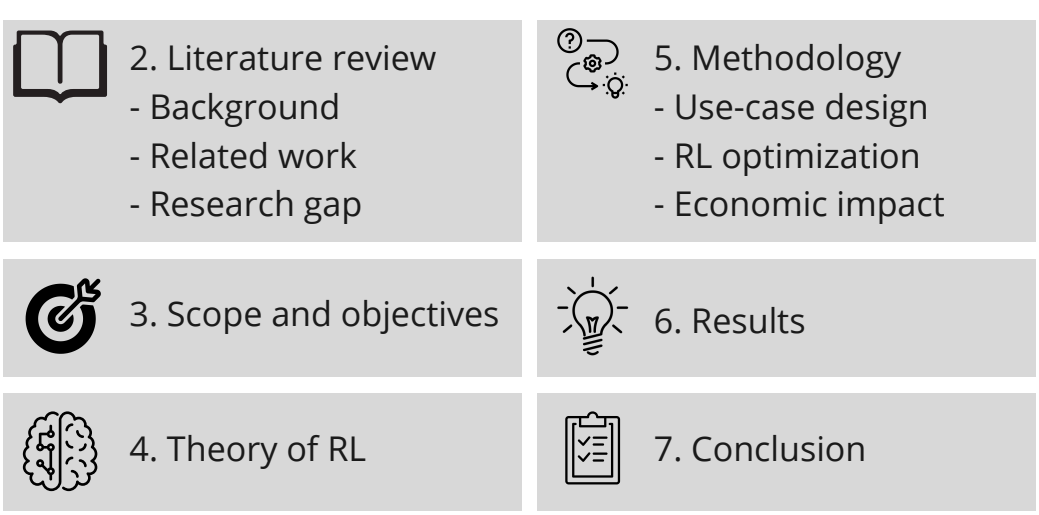


Figure 1.0.1: Thesis structure

Chapter 2

Literature review

2.1 Background

In this section, a background is provided on the German Electric Vehicle (EV) landscape with its trends, regulations, and subsidy schemes. Subsequently, the most common optimization approaches for EV charging management are presented.

2.1.1 The German EV landscape

Germany has seen a significant increase in its EV stock since 2018, as can be seen in Figure 2.1.1.

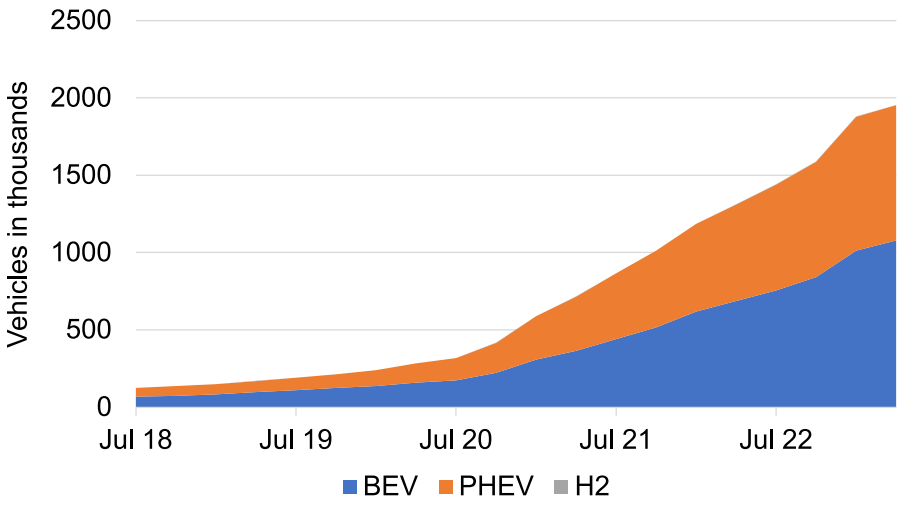


Figure 2.1.1: EVs in Germany [10]

As of July 2023, ca. 1.1 million EVs and 876 thousand Plug-in Hybrid Electric Vehicles

(PHEVs) are in circulation, corresponding to ca. 80 GWh of battery capacity [10, 11]. The cumulative onboard charger power sums up to ca. 114 GW of DC charging power, and ca. 14 GW of AC charging power [10]. To put this into perspective, the German peak load only amounts to ca. 80 GW [12].

Alongside EVs, charging stations have been steadily deployed as well: As of May 2023, 36800 public charging stations have been registered by the Federal Grid Agency - an increase from 25900 by the end of 2021 [13, 14]. An official estimate of the total number of chargers in Germany revolves around 65000 [15]. A detailed account of public chargers, their usage, and profitability can be found in [14]. When it comes to commercial applications of EVs, a similar trend can be observed [11]. EVs in commercial applications are further elaborated upon in Section 5.1.

2.1.2 Regulatory measures

On an EU-level, the Alternative Fuels Infrastructure Directive mandates its member countries to make electric mobility possible for passenger, light-commercial, and heavy-duty vehicles. A focus is set on interoperability to allow for seamless operation between member countries - this is enacted by the German regulation on charging infrastructure, which defines the technological standard for charging stations in Germany [16]. Further, economic incentives are put in place to accelerate the transition: when buying an EV, up to €6750 can be deducted from the purchase price via subsidies [15, 17].

Additionally, tax incentives are put in place: no tax is to be paid when charging an EV at the employer [18]. For companies, a support scheme existed until 2022 that alleviated 70% of capex for investments in charging infrastructure, up to €45000 [19]. Note that apart from the national level, each state has the possibility to enact its own subsidy schemes to support electric mobility, e.g. [20].

Regulatory measures also exist to advance in the digital transformation of the energy sector: in May 2023, a law was passed on Smart Meters, mandating their installation for households with an energy consumption above 6000 kWh or PV above 7 kWp [21]. Germany currently lags behind by a considerable margin when it comes to smart meter installations. The passing of this law therefore, sets the necessary foundation for load management business models that rely on detailed information and communication regarding energy demand.

2.1.3 EV charging management

Although the infrastructure for EV charging management is still under development in Germany, multiple business models are emerging, ranging from dynamic electricity tariffs to dynamic load management.

On the supply side, novel business models can be found regarding time-of-use tariffs and dynamic electricity pricing. These companies usually enable their customers to leverage intraday price differences on the EPEX spot market. The value proposition is accompanied by empowering the end-user to make better decisions on consumption timing via informative dashboards and data integration. Two commercially available examples can be found in [22, 23].

On the demand side, dynamic load management solutions are experiencing a boom. The core value proposition usually consists of the ability to optimize the EV chargers or other assets (e.g. heat pumps, PV, appliances), achieving a cost reduction compared to the status quo. Additional value propositions include consulting services (e.g. charging infrastructure design), or the seamless integration of software and hardware. Players with a high number of compatible chargers and communication protocols thereby set each other apart from the competition. Dynamic load management is offered by start-ups, EV supply equipment (EVSE) manufacturers, utilities, and other players in the domain of digital energy solutions - the level of competition is therefore rather high. Commercial examples can be found in [24–27].

2.1.4 Hardware and software requirements

Implementing dynamic load management in the real world requires metering, communication, and control. On the most fundamental level, metering could consist of gathering information on the power demand of the EV chargers, thereby ignoring building and other appliances. In this case, a static limit would be set that the chargers must not exceed. The second option is a smart meter that provides information on the building load, PV, or other individually connected appliances with the required communication channels (e.g., heat pump, smart fridge, etc.). A smart meter provides the advantage that the controller can dynamically adjust according to the current load of the building, allowing for more optimization potential. The second technical prerequisite is communication. Signals must be sent and received by the load

controller in real time in order to take the right actions at the right time. The most commonly used protocols are the Open Charge Point Protocoll (OCPP), ISO 15118, ModBus TCP, EEBus, and MQTT [28–31]. Taking the example of OCPP, information on the State of Charge (SOC), time of arrival, type of vehicle, and other parameters are exchanged. Finally, a load controller is required to perform computation and to send the scheduling signals to the EV charger. Depending on the scope of the optimization, the computational complexity can vary considerably. When researching hardware components for this study, it was found that several controllers were based on a Raspberry Pi platform, or similar architectures [24, 32, 33].

2.1.5 EV charging optimization methods

With increasing electrification of transport in domestic, as well as commercial sectors, uncoordinated charging is starting to pose a challenge: it can increase peak loads, cause grid congestions, or affect power quality due to voltage deviations [34, 35]. Regulators and utilities are therefore actively shaping the EV charging landscape to accommodate for this transition. For example, electricity tariffs are being put in place that incentivize postponing the charging process until the night hours [36]. With dynamic electricity tariffs, an economic incentive has emerged to optimize the charging process. The problem of minimizing charging cost while respecting the mobility constraints has therefore been addressed with a multitude of approaches. This section gives a brief overview of the most prominent methodologies.

Linear Programming (LP) and Mixed Integer Linear Programming (MILP) are among the most commonly used techniques, offering a structured approach to optimizing the charging schedule based on a set of linear constraints and objectives [5–7]. Quadratic and Non-linear programming methods extend these techniques to handle more complex, non-linear relationships, such as battery degradation [34, 37]. With their mathematical frameworks, LP-techniques provide a deterministic approach, and find the optimal solution within the given problem formulation. However, LP-methods do not perform well on problems with uncertainty, as well as dynamic environments [8, 38].

Genetic algorithms, particle-swarm optimization, and other meta-heuristic methods have also been applied to EV charging optimization, offering a more exploratory approach that can potentially uncover novel solutions [39–42]. These methods operate

by iteratively generating and refining potential candidates based on a fitness function, which could represent the charging cost in the context of EV charging. While meta-heuristic methods can perform well in non-linear, or uncertain environments, they often require significant computational resources and may not always converge to the optimal solution [42].

Due to the relevance of EV charging management and the shortcomings of the different optimization approaches, new methods have been developed and tested continuously. One of these methods is RL, which models the EV charging problem as a sequential decision-making process with penalties and rewards after each decision. RL has emerged in the EV charging space due to its suitability in uncertain and dynamic environments, as well as its low computational intensity after the training phase [38]. RL-based EV charging can therefore fulfil the high requirements for real-time optimization that emerge when instantaneous reactions are required in dynamic, uncertain, or non-linear optimization problems. Attention particularly surged in recent years, after RL achieved breakthroughs in various fields.

2.1.6 Breakthroughs of RL

Reinforcement learning first attracted significant attention in 1994, when an agent called TD-Gammon was successfully trained to compete with world-class players in backgammon, a game with ca. 10^{20} possible states [43]. A model-free approach was chosen, meaning that the understanding of the game's rules and an optimal strategy were developed through experiencing possible outcomes of the game, and without prior knowledge. TD-Gammon consisted of a neural network with MLP architecture and one hidden layer [43]. The neural network was used to approximate the value function. Based on the value function, a policy was chosen that could always reach the state with the highest value. At that time, neural networks were already identified as powerful tools, able to approximate any non-linear function given a sufficient number of hidden units. Soon after the initial encouraging results, however, research suggested that problems of divergence can occur for a general set of problems [44, 45]. Instability was particularly observed for frameworks that featured the so-called "deadly triad": function approximation, bootstrapping and off-policy learning [46]. This resulted in a period of caution regarding research in the field of RL with function approximation and Deep Reinforcement Learning (DRL) [47]. In the 2000s, a combination of

computational performance increases and new algorithm designs partially alleviated the problems of the deadly triad [47].

Interest and attention were re-ignited, when DeepMind combined the emerging field of deep neural networks with function approximation-based reinforcement learning: an agent was presented that, by learning a general policy, outperformed previous algorithms and even humans on a variety of Atari games [48, 49]. The proposed approach, deep Q-learning / DQN, processed raw pixels as input and returned a set of discrete actions (up to 14) that represent the controls of the Atari joystick. Two years later, the game of Go was mastered by a DeepMind RL algorithm [50]. In an award-winning documentary about the algorithm, the DeepMind developers can be seen rooting for the human world champion Lee Sedol to at least win one game - and thus prove that humanity still stands a chance against artificial intelligence. He ended up losing 4-1.

Today, RL is used in different areas in the real world, such as traffic control, autonomous driving, finance, healthcare, gaming, and robotics [51]. Research efforts are being dedicated to making these applications safe and robust, because the aforementioned fields involve interactions with other agents, as well as human beings.

2.2 Related work

Applications of RL in the domain of EV charging have grown exponentially over the past ten years [52]. This can be attributed to the promising results that were already achieved during the early stages of research, combined with the relevance of EV charging management in the energy transition. Progress in the fields of RL has led to the deployment of increasingly sophisticated algorithms and models that aim to recreate real-world scenarios.

In 2013, a tabular representation of a domestic, unidirectional EV charging problem was proposed [53, 54]. Statistical data was used to create realistic mobility schedules, and real-world data was used for household consumption. The goal of the agent was to maximize the total welfare of the household, which comprised household energy consumption and the charging amount of the EV. A Q-Learning agent was chosen which maximized the state-action value $Q(s, a)$ for every time step. Constraints were

applied to ensure a minimum level of mobility, and to cap the maximum charging power according to grid limitations. The results showed a maximum cost reduction of 30% and a maximum peak reduction of 25%. The exploration of V2G and social welfare were identified as future improvements. The work of [53, 54] showed that promising results can be achieved with a discrete Q-learning approach - a relatively basic architecture compared to the current state of the art.

Although the results in [53, 54] were promising, they also featured one main drawback: the observations and actions were discrete in order to solve the problem in a tabular representation. Tabular approaches can theoretically find the global optimum of a decision problem due to the finite nature of the problem [47]. However, tables can become prohibitively large and complex, up to the point where a tabular representation is no longer feasible [8]. Regarding EV charging, this can arise when multiple vehicles are considered, alongside other parameters such as price, load, PV, etc. This problem is known as the curse of dimensionality: problems with small dimensions are often too simple, whereas realistic approaches feature prohibitively large dimensions with infeasible computational complexity [46]. Solutions to this problem come in the form of state space reduction and function approximation of the tabular representation.

Both approaches were combined in [55]: a novel approach for EV charging was proposed, that does not scale with the number of charging points. This was achieved by grouping EVs with the same properties regarding required charging time and remaining time until departure. The resulting state space representation only scaled with the maximum charging duration and the granularity of the time steps. In the model, the objective was to flatten the load of EV charging. A Q-Learning algorithm was chosen that approximated the Q-function via a Deep Neural Network (DNN). Additionally, data was fed into the RL agent in batches of randomly generated episodes. This approach is known as batch RL with fitted Q-iteration. Since a DNN is used in this case to approximate the Q-function, the approach falls under DRL. The framework proved to be a feasible representation that can be applied to different fleet sizes without differences in performance. However, a low granularity of 2 hours was chosen to reduce the state space and the episodes were terminated at midnight. Real-time control and charging in the early morning hours to satisfy the demand of the previous day were therefore not possible. Compared to business as usual, a load flattening of 37% could be achieved. The identified next steps included exploring the trade-off

between granularity and model complexity and validating the framework in a real-world scenario.

A thorough literature review of RL applications in EV charging optimization was performed in [8]. More than 50 papers were reviewed that applied RL to optimize the charging process, with publications dates ranging from 2011 to 2021. According to Abdullah et al., the predominant optimization objectives were cost minimization and profit maximization, depending on the perspective of the study. In almost all studies reviewed, the state space contained at least the following two points of information: charging demand, and time left at the charger. Additional state variables consisted of charging costs, weather data, renewable generation and electricity loads. The most used state representation was the combination of SOC, electricity price and time left at the charger. It was also found that the implementation of constraints can be useful and even necessary in some cases because constraining the agent's actions can be an effective way to include knowledge of the real system in the RL environment. The most frequently implemented constraint was the charging network limit, i.e. power capacity or transformer loading limits, which helped to prevent the agent from learning unrealistic scenarios. As for the implemented methods of RL, a transition from Q-Learning to DRL was observed. In Figure 2.2.1, the transition is illustrated. Years 2011 and 2021 only make up three data points and were thus omitted.

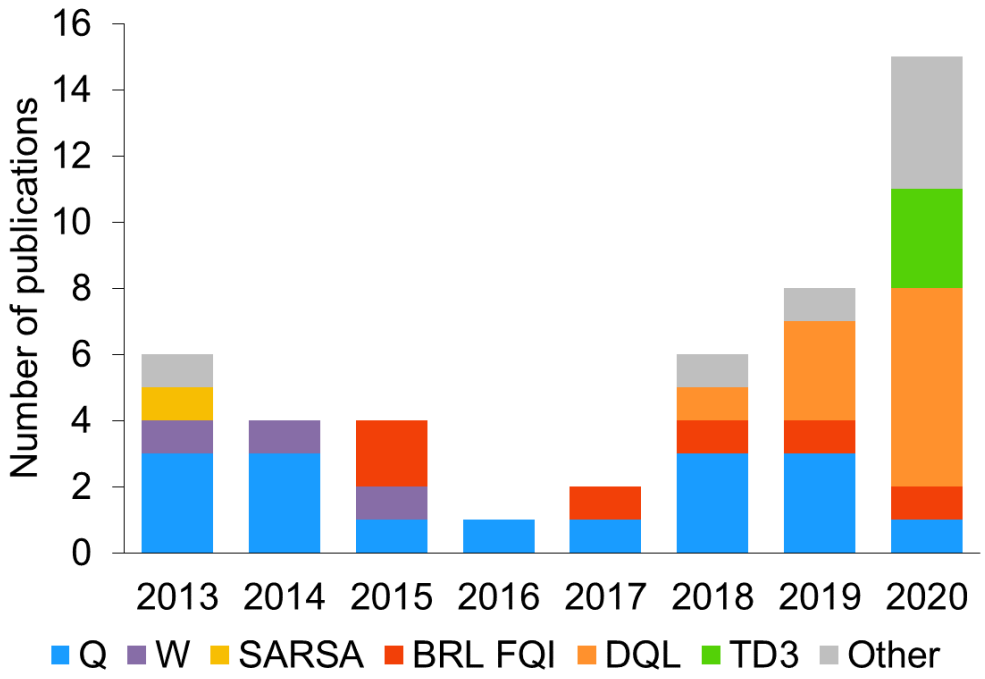


Figure 2.2.1: RL methods used from 2013 to 2020 according to [8]

From 2018 onwards, DRL methods such as Batch-RL with fitted Q-iteration (BRL FQI) with DNN-based approximation, Deep Q-learning (DQL), and Twin Delayed Deep Deterministic Policy Gradient (TD3) made up a significant portion of scientific contributions.

The main takeaways from the paper are [8]:

1. The studied RL methods outperformed their benchmarks, such as uncontrolled charging or other rule-based approaches.

2. SOC and time left at the charger are essential for the state representation.

3. Constraints on the charging infrastructure can be useful for realistic behaviour.

4. DRL methods allow for a more realistic and efficient optimization and have established as the state-of-the-art for RL-based EV charging optimization.

Abdullah et al. also discussed current research limitations. Although less computationally intensive than conventional optimization techniques, RL still requires considerable computational resources, which must be taken into account when designing real-world applications. Fog computing, a combination of cloud and edge computing, was proposed as a promising architecture for RL-based charging management systems [8]. Additionally, some models still rely on perfect knowledge of the vehicles' schedules, or take other simplifying assumptions. The authors thus highlighted the research gap in modelling more realistic scenarios and applying RL frameworks in the real world.

Similarly, a critical review of RL applications in the EV charging domain was performed in [38]. The authors analysed 54 articles and sorted them by use-case: G2V (unidirectional charging), V2H, V2G (arbitrage), and V2G (FCR). Similar to [8], the chosen representations for state, action, and reward, as well as the chosen RL algorithm were analysed. The chosen RL algorithms are shown below, sorted by use-case. The publications range from 2015 to 2022, with the large majority being published between 2019 and 2022.

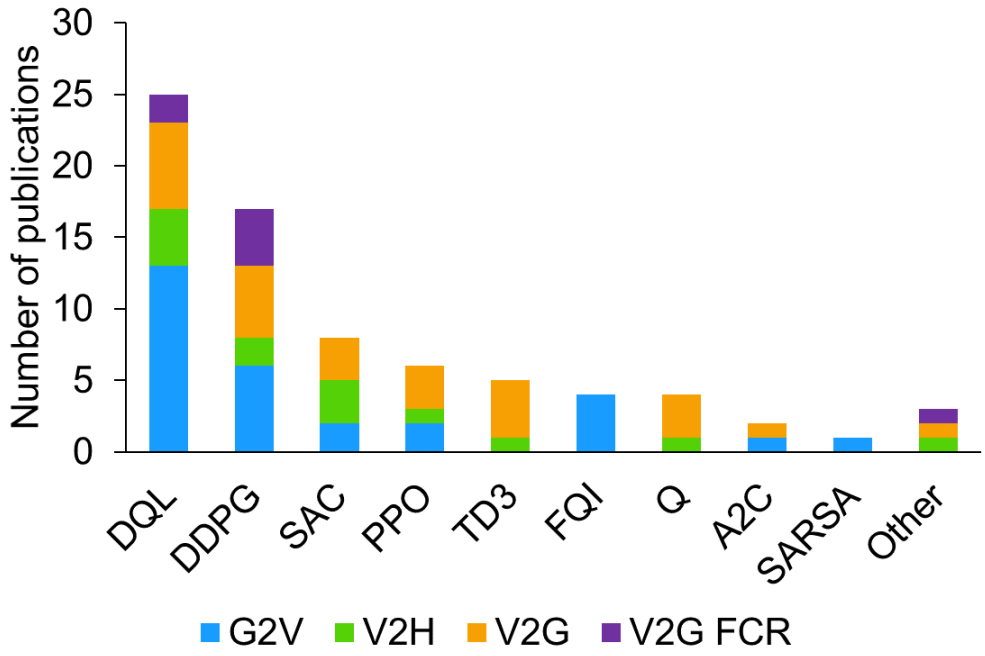


Figure 2.2.2: Chosen RL methods by use-case according to [38]

As can be seen, DRL methods (the first five in the graph) are predominantly used among all use-cases, indicating that the complex EV environment requires DRL methods. The authors highlighted that real-time communication is paramount for V2H and V2G and that delays or data loss could severely affect the functionality of the service. It has not been studied how real-world communication affects the RL environment [38]. Another research gap was identified for RL-based V2G implementations, where only a few publications exist. For data privacy reasons, Multi-Agent Reinforcement Learning (MARL) was proposed instead of centralized, single agent RL architectures.

2.3 Summary and research gap

A review of the German EV landscape revealed that the boom of EVs in Germany is posing challenges that the grid was not designed for. Peak and uncontrollable loads are increasingly introducing strains, as well as voltage and frequency fluctuations. Flexibility measures on both the supply and demand side are therefore moving into the spotlight of the energy transition, and the need for managing the EV charging process is becoming paramount. Additionally, there is an equally high demand for EV charging optimization from the commercial perspective, due to economic and up-time

incentives: Overall charging costs are to be minimized, and real-time peak shaving is required to keep the kW_{peak} quarter of the year at its lowest possible value. Real-time capable algorithms are therefore gaining in importance.

Linear Programming, a commonly used technique to solve optimization problems, performs well if the problem is linear, all constraints are known, and perfect information is available. These requirements are often not fulfilled in the real-world, where probabilistic, non-linear, and uncertain events occur on a regular basis [8]. Further, the LP model needs to be rebuilt every time a change occurs in the real-world representation of the problem. Where linear models fall short, non-linear, meta-heuristic or model-predictive control methods can alleviate some limitations. However, computational intensity limits the real-world applicability of these methods. In literature, a recognized need for RL-based optimization has therefore been established in domains such as EV charging, because RL is capable of dealing with non-linear, probabilistic, and uncertain problems in real-time. The performance of RL-based methods drastically increased in recent years, and RL algorithms now outperform previous methods, as well as human beings in a variety of domains. It thus becomes evident that there is a need for optimization of EV charging processes with RL.

Reviewing literature on RL-based EV charging yielded the following research gaps:

1. EV charging environments for RL exist, but are not realistic enough. Episodes are either terminated in between days, or other simplifications are taken (hard-coded price curves, no battery degradation, low time resolution), which reduce the real-life applicability. No framework was found for commercial use-cases.
2. When RL is used to optimize EV charging, the focus is often purely technical and focused on the reward signal: economic calculations do exist, but rarely extend beyond percentage savings. Investment calculations that take into account capex (e.g., load controller) and third party fees regarding dynamic load management were not found during this literature review.

The needs and research gaps helped to focus the efforts of this study: modelling three realistic commercial use-cases as a Markov Decision Process (MDP), and optimizing them using RL. A novel framework for RL-based EV charging is implemented in Python that is tailored to commercial applications, and that avoids simplifying

assumptions that were found in literature. Apart from benchmarking the agents with rule-based strategies, a linear optimization is built to compare the agent with a deterministic optimization that possesses perfect knowledge. The scope and objectives will be more closely explained in the next chapter. Figure 2.3.1 summarizes the identified needs and research gaps.

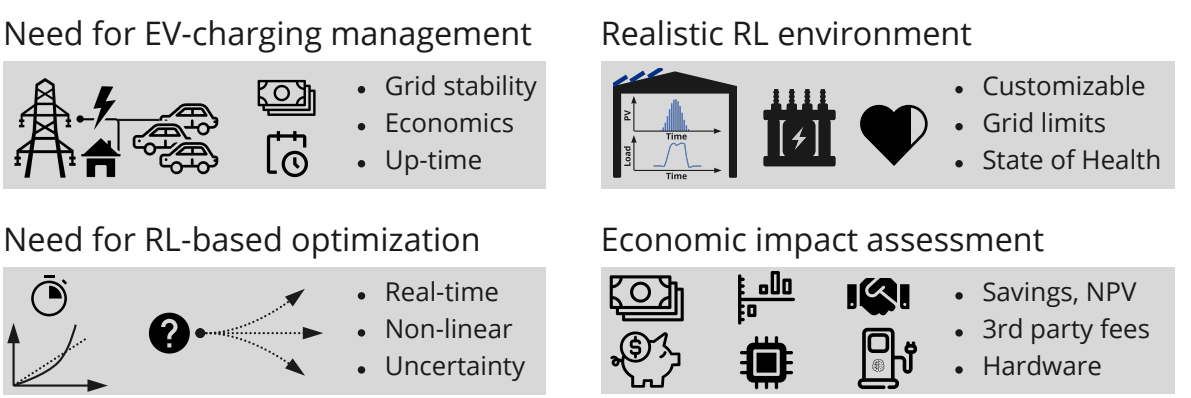


Figure 2.3.1: Identified needs and research gaps

Chapter 3

Scope and objectives

The literature review revealed a need for EV charging optimization with RL due to the non-linear and probabilistic nature of the problem, as well as real-time requirements that emerge in real-world applications. Further, a research gap was identified for a realistic study on commercial use-cases that avoids simplifying modelling assumptions, and takes into account the required investments and expenses to set up a smart-charging business model. This study addresses the aforementioned research gap and adds to the research space of EV charging with a generalizable approach of solving realistic EV charging use-cases with RL.

This study will optimize the EV charging schedules of three commercial use-cases using Reinforcement Learning (RL). The performance of RL will be analysed quantitatively: first, by benchmarking the results with commonly implemented charging strategies, and second, by comparing the results of the three use-cases, revealing whether smart-charging suitability differs across commercial domains. A net-present value analysis supports the comparison by considering investment in load controlling and communication infrastructure, as well as third-party fees for providing the load management service.

Two overarching research questions motivate the study and tie in the research efforts:

1. *What is the potential of RL-based EV charging in terms of cost savings and real-world applicability when compared to the most common EV charging strategies (e.g. uncontrolled charging and optimization-based smart charging)?*

2. *How do the three investigated commercial use-cases differ regarding their cost savings and potential to leverage RL-based charging optimization?*

The geographical scope of this study is Germany. As for the commercial use-cases, the following will be analysed:

- A utility that operates a fleet of service vehicles to conduct maintenance on their infrastructure or at the customer, e.g. Vattenfall, E.ON.
- A last-mile delivery company delivering parcels or goods from the distribution centre to the customer, e.g. DHL, ICA.
- A caretaker company that operates a fleet of vehicles to service their patients, e.g. Malteser.

Two modes of operation will be analysed for the RL agent and its benchmarks. First, an arbitrage scenario is designed in which the agent is able to trade energy on the spot market, allowing complete exploitation of intraday price differences, bidirectionally. Second, a realistic scenario is designed: electricity tax, grid fees and surcharges are added to the electricity price. For discharging, the agent does not receive the spot price, but the current German PV feed-in tariff minus third-party fees. Vehicle-to-grid in form of frequency regulation is not implemented in this study.

Efforts were made to develop a generalizable framework that can be applied to other use-cases, both geographically, and commercially. Choices regarding the Python implementation and data sourcing were made with these efforts in mind.

Chapter 4

Theory of Reinforcement Learning

This chapter covers the theoretical prerequisites to understand the applications of RL in EV charging. First, the big picture of RL will be introduced, followed by the theoretical foundations of RL. Finally, the chosen Deep Reinforcement Learning algorithms Twin Delayed Deep Deterministic Policy Gradient (TD3) and Proximal Policy Optimization (PPO) will be explained in detail.

4.1 The big picture of RL

RL methods can be divided into two main categories: model-based RL and model-free RL. Model-based RL frameworks are based on a model of the world that is either provided or learnt during the training process. The understanding of the world is then used to infer the best possible action. Model-free RL, on the other hand, relies on no such information. Instead, the RL agent learns through training how good a certain state is (value-based), or how good a certain policy is (policy-based). No knowledge of the world's dynamics is required in this approach. Nevertheless, the agent must make sufficient observations of the world during the training process to make rational decisions.

Further, RL frameworks can use Deep Neural Networks to approximate the value function or policy, in which case the method falls under the category of Deep Reinforcement Learning. If multiple agents are used, the method is referred to as Multi-Agent Reinforcement Learning (MARL). An overview is provided in Figure 4.1.1. Note that the methods shown can both apply to RL and MARL. Detailed explanations

of the mentioned algorithms can be found in [38, 46].

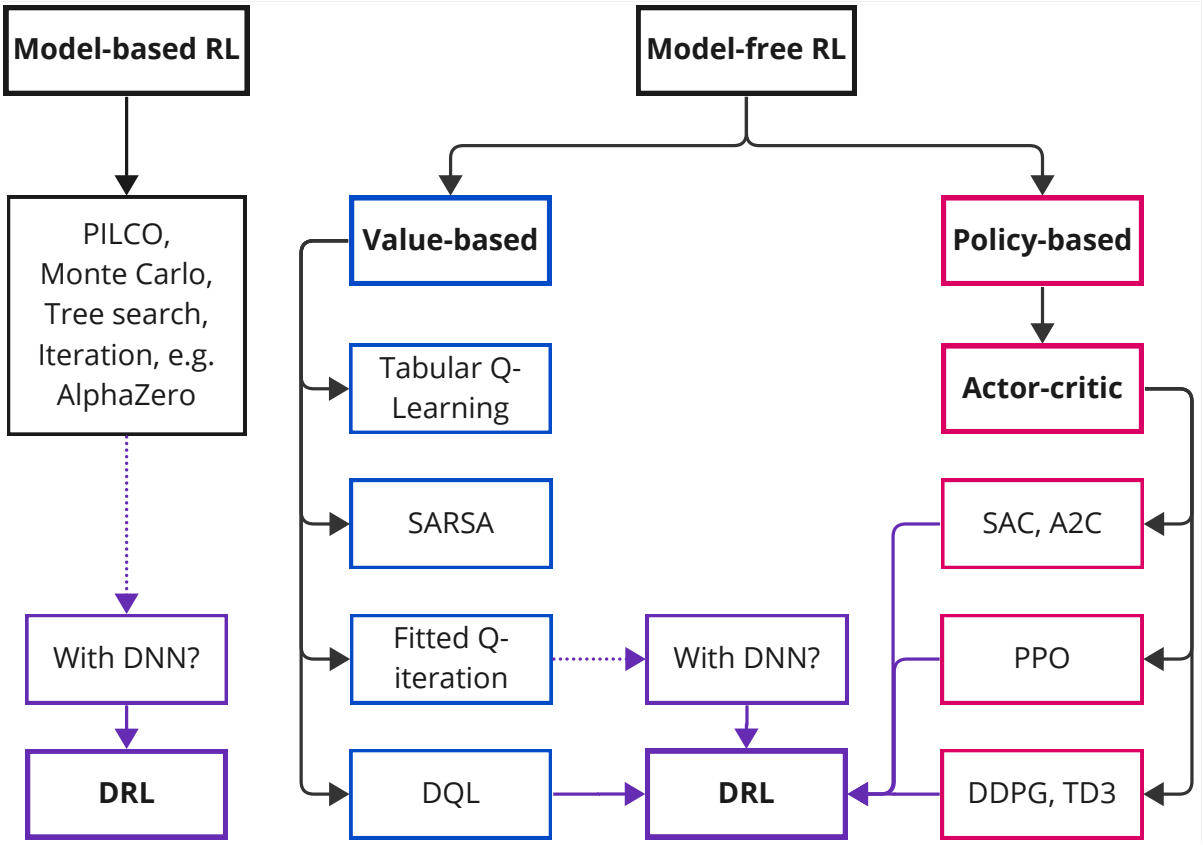


Figure 4.1.1: Overview of RL methods [38] [1]

Each algorithm has its own use-case. The main distinguishing property is the ability to process discrete or continuous information. To be able to apply RL effectively in real-world problems, it is crucial to understand both the problems' environment and action space, as well as the capability of the algorithm to deal with the respective data. Both the observation (the input into the RL agent) and the action (the agent's output) can be either continuous or discrete. Additionally, the value function can either be represented by a table or by an approximation function, such as a linear approximation or a DNN. Table 4.1.2 summarizes these key characteristics of the most important RL algorithms. Casting thereby refers to the fact that a discrete action space can be obtained by casting continuous variables to discrete values. One example for this could be rounding to the nearest integer or decimal point.

[1]The purple rectangles serve as an aid to distinguish between tabular RL and deep RL methods.

Algorithm	Discrete State Space	Continuous State Space	Discrete Action Space	Continuous Action Space	Value Function Representation
Tabular Q-Learning					Tabular
SARSA					Tabular
W					Tabular
Fitted Q-Iteration		DNN needed			Linear or DNN approximation
DQL					DNN
SAC			Casting		DNN
DDPG			Casting		DNN
TD3			Casting		DNN
PPO			Casting		DNN
A2C			Casting		DNN

Figure 4.1.2: RL algorithms and their characteristics [56]

4.2 RL fundamentals

RL is a computational approach to learning via interaction with an environment. It is a field of machine learning that can tackle complex problems by modelling them as decision-making processes. In RL, an agent interacts with an environment and seeks to achieve the optimal outcome by taking the right actions at the right time. Just like in the real world, there can be uncertainty or delayed consequences that require strategic behaviour. Trade-offs might have to be made between an immediate reward and a potential reward in the future. When an agent is first introduced to its environment, it does not possess the knowledge to evaluate its actions. It must therefore explore and learn through trial and error, as well as understand that some actions might only be beneficial at a later point in time [46].

4.2.1 Markov decision process

RL is considered as a third Machine Learning (ML) paradigm, next to supervised and unsupervised ML. While supervised ML revolves around predictions from labelled training data, and unsupervised ML aims to find patterns in unlabelled data, RL aims to maximize the reward signal in a sequential decision-making process [8, 46]. The

sequential decision-making process used in RL is known as a Markov Decision Process (MDP), which consists of an agent and an environment that interact with each other via states, actions, and rewards, as shown below [57].

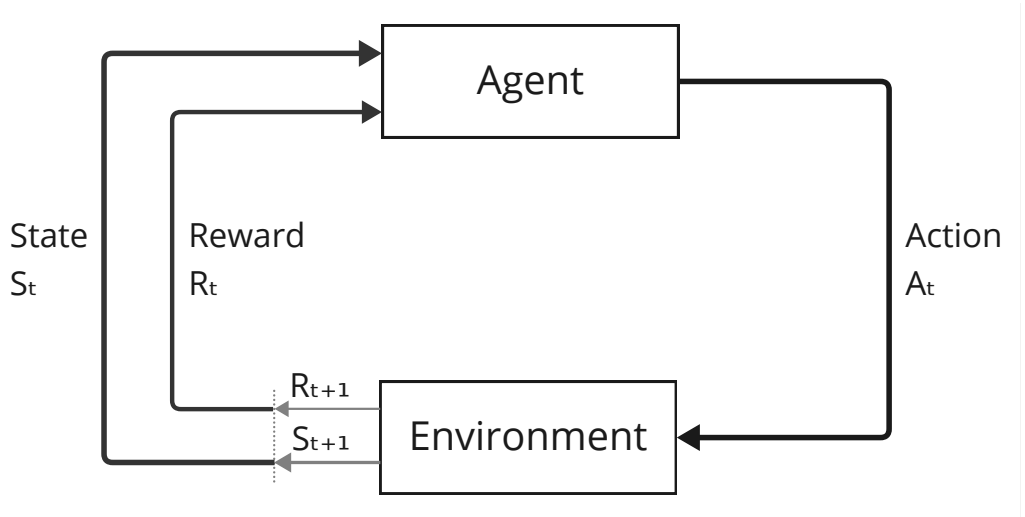


Figure 4.2.1: Markov decision process [46]

The MDP is a closed loop system with discrete time steps. At any time step, the agent receives a representation of the environment $S_t \in S$, and selects an action $A_t \in A$. In the following time step, the agent receives a rewarding or penalizing signal $R_{t+1} \in R$ from the environment. Over time, a trajectory forms: $S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, \ldots$ This trajectory can be either finite or infinite. Most mathematical frameworks revolve around the finite MDP, but the key ideas are also applicable in a continuous context [46].

In the finite MDP, the state and reward at time step t, S_t and R_t, can be described via probability distributions. In an MDP, the values of the next state and reward $s' \in S$ and $r \in R$ only depend on the preceding state and action, as shown below:

$$p(s', r|s, a) = P(S_t = s', R_t = r|S_{t-1} = s, A_{t-1} = a) \tag{4.1}$$

The assumption, that s' and r only depend on the preceding s and a is known as the Markov assumption. A problem is Markov, when the entire information of the environment is encoded in the previous state and action. If the assumption is violated, it is not possible to properly match the right action to the right state, because two states can appear similar even though they are entirely different.

4.2.2 Returns and discounting

In RL, an agent's goal is to maximize the expected cumulative reward. This can be explained intuitively: For example, in chess the objective is to win the game instead of making one or several good moves. At any given point in time t, the reward sum can be written as the return $G_t = R_{t+1} + R_{t+2} + R_{t+3} + ... + R_T$, with T being the final time step. A final time step exists in problems that either have a termination condition or can be broken down into different episodes. This is the case for most computer games, and many real-world problems. Continuous control problems, however, do not necessarily terminate and have an indefinite time horizon. A sum of rewards would go towards infinity, which is why a discounting factor γ is introduced to the equation. An agent with a discounting factor of zero would only value immediate return, whereas an agent with a discounting factor of one would have no time preference [46].

$$G_t = R_{t+1} + \gamma \cdot R_{t+2} + \gamma^2 \cdot R_{t+3} + ... = \sum_{k=0}^{\infty} \gamma^k \cdot R_{t+k+1} \tag{4.2}$$

For an infinite time horizon, γ is defined within the half-open interval between zero and one: $\gamma : [0, 1)$. This way, the sum of discount factors converges as shown in Eq. 4.3.

$$\sum_{k=0}^{\infty} \gamma^k = \frac{1}{1-\gamma} \tag{4.3}$$

4.2.3 Policies and values

The agent maximizes the expected return by deciding which action to take for a given state. This decision takes place in the so-called policy function. The policy function is a mapping from states to actions:

$$\pi : s \to a = \pi(a|s) = P(A = a|S = s) \tag{4.4}$$

where π is the conditional probability distribution of $a \in A$ given $s \in S$.

Policies are evaluated by calculating the expected return when $\pi(s|a)$ is followed. There are two ways to calculate the expected return: via the state-value function and via the

action-value function.

The state-value function $v_\pi(s)$ is the expected return under policy π, when starting in state s:

$$v_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s\right], \forall s \in S \tag{4.5}$$

The action-value function $q_\pi(s, a)$, on the other hand, is the expected return under policy π, when starting with the state-action pair (s, a):

$$q_\pi(s, a) = \mathbb{E}_\pi[G_t|S_t = s, A_t = a] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s, A_t = a\right], \forall s \in S, a \in A \tag{4.6}$$

Computing average returns for each state would eventually converge to the state-value function $v_\pi(s)$, whereas computing average returns for each separate state-action pair would eventually yield the action-value function $q_\pi(s, a)$. The functions $v_\pi(s)$ and $q_\pi(s, a)$ are related to each other: summing $q_\pi(s, a)$ over all actions for a certain state yields $v_\pi(s)$, as shown in 4.7.

$$v_\pi(s) = \sum_a \pi(a|s) q_\pi(s, a) \tag{4.7}$$

Additionally, both v_π and q_π are related to their successor states as shown in Eq. 4.8 and 4.9. These fundamental relationships are known as the Bellman backups for v_π and q_π [58].

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_\pi\left[G_t|S_t = s\right] = \mathbb{E}_\pi\left[R_{t+1} + \gamma v_\pi(S_{t+1})|S_t = s\right] \\ &= \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)\left[r + \gamma v_\pi(s')\right], \forall s \in S \end{aligned} \tag{4.8}$$

$$\begin{aligned} q_\pi(s, a) &= \mathbb{E}_\pi\left[G_t|S_t = s, A_t = a\right] = \mathbb{E}_\pi\left[R_{t+1} + \gamma q_\pi(S_{t+1}, a')|S_t = s, A_t = a\right] \\ &= \sum_{s',r} p(s', r|s, a)\left(\sum_{a'} \pi(a'|s')\left[r + \gamma q_\pi(s', a')\right]\right), \forall s \in S, a \in A \end{aligned} \tag{4.9}$$

where s' is the next state, and a' the next action.

4.2.4 Dynamic Programming and optimal policies

To achieve maximum returns, the agent must find an optimal policy π_* that maximizes the value function. This can be done with Dynamic Programming: a set of algorithms designed for this purpose. To highlight one method, value iteration, will be explained [46].

Value Iteration

1. Policy evaluation: Starting with an arbitrary policy π, its state-value function is determined for all states according to Eq. 4.8.
2. Policy improvement: For each state, the policy is updated to π', taking the action that yields the highest value: $v_{\pi'}(s) = \max_a \sum_{s',r} p(s', r|s, a) [r + \gamma v_{\pi'}(s')]$
3. This process is then repeated until $\pi \approx \pi_*$ such that:

$$\pi_* = \arg\max_a \mathbb{E}_\pi [R_{t+1} + \gamma v_\pi(S_{t+1})|S_t = s] = \arg\max_a \sum_{s',r} p(s', r|s, a)[r + \gamma V(s')]$$

This is shown in the example below. Figure 4.2.2 shows a grid world, where four directions of movement are possible, except for the borders. When moving into a grey field, the position remains unchanged. Some fields return a positive or negative reward. In the beginning, an arbitrary policy is initialized, where the agent remains on its field or moves along the edges. This policy is evaluated in the first figure. In a subsequent step, the policy is improved, which can be seen by the change of arrows. In Sub-figure (c), the updated policy is evaluated once more. Taking the example of the lower green square with the red border, the following calculation takes place. Gamma, the discounting factor, is chosen to be 0.9 [59].

$$v_\pi(s) = \mathbb{E} [R_t + \gamma v_\pi(S_{t+1})] = 0 + 0.9 \cdot 1.0 = 0.9 \tag{4.10}$$

Sub-figure (f) shows the converged value function, as well as the optimal policy: for each field, it is shown which direction yields the highest value. After a sufficient number of iterations, the information of the green field (reward = 1) has propagated back to the very edges of the board, and the best action is known for each square.

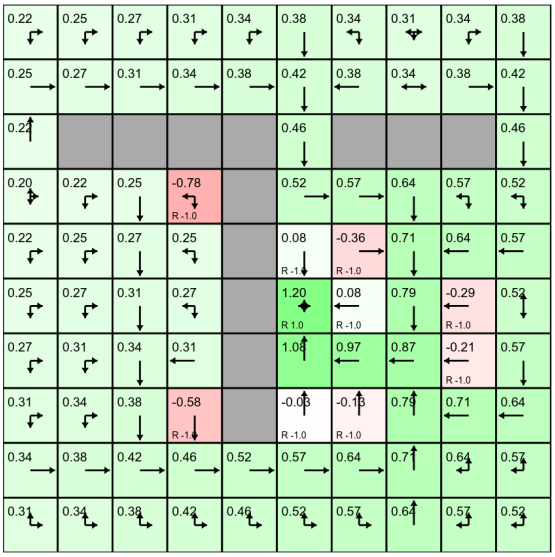
(a) First policy evaluation

(b) First policy improvement

(c) Second policy evaluation

(d) Second policy improvement

(e) Third policy evaluation



(f) Optimal policy

Figure 4.2.2: Value iteration in a grid world environment [59]

4.2.5 Exploitation and exploration

In the example above, the policy was chosen to always maximize value; this is known as a greedy policy. If an RL agent always chooses the action it believes to be the best, unknown or unforeseen circumstances might not be taken into account because they are never experienced. A trade-off therefore exists between exploiting the current knowledge and exploring new states. The trade-off between exploration and exploitation is a fundamental feature (and problem) in RL. One way to incorporate exploration is to take a random action with probability ϵ. This is called an ϵ-greedy policy.

$$\pi(s) = \begin{cases} \arg\max_a q_\pi(s,a) & , 1-\epsilon \\ rand(a) & , \epsilon \end{cases} \tag{4.11}$$

By adding an element of randomness to the agent's actions, the agent experiences a wider variety of states. This can be particularly useful if environments change over time. In such a case, an ϵ-greedy policy would rediscover a changing optimum eventually, whereas the same would not be guaranteed without exploration.

4.2.6 Monte Carlo and Temporal-Difference Learning

Dynamic programming, as shown in Section 4.2.4, is a feasible approach, when a perfect model of the environment is embedded in the RL agent (model-based RL). In this case, the knowledge of which state to go to next suffices, because the agent knows which action to take to transition to the next state (i.e. in the example above). In many cases, however, the agent does not possess a deterministic model of the world (model-free RL), and requires information on which action is best in a certain state. Transitions of the MDP must then be sampled to build an understanding of the underlying problem, which can be accomplished by Monte Carlo and Temporal-Difference Learning, for example.

Monte Carlo

Monte Carlo methods gain an understanding of the problem by sampling complete episodes and averaging returns for each state. The premise of MC methods is that by averaging the returns for each state over many episodes, the law of large numbers

guarantees that the estimates will converge to their expected values. If G_t^1 is the return following the first occurrence of state s in an episode, the value $V(s)$ is estimated as the average return from state s after many episodes and visits to that state:

$$V(s) = \frac{1}{N(s)} \sum_{n=1}^{N(s)} G_t^i \tag{4.12}$$

where $N(s)$ is the number of times state s is visited and $G_t^{(i)}$ is the return following the i-th visit to state s. MC methods require samples of complete episodes and cannot be computed while the episode is still ongoing. While the approach is intuitive and conceptually simple, it can require a large number of episodes to converge, particularly in complex environments with large state and action spaces [49]. Nonetheless, MC methods provide a powerful and flexible tool for model-free RL when the task is episodic and the environment's dynamics are unknown or too complex to model accurately.

Temporal Difference learning

Temporal Difference (TD)-Learning is a central method in the realm of model-free RL as it combines ideas from both dynamic programming and Monte Carlo methods. Similar to Dynamic Programming (DP), TD uses estimates of successor states. Similar to Monte Carlo (MC), TD samples from the environment while experiencing it. TD-Learning estimates the value function through iterative updates within an episode, and does not require an episode to be completed. After each time step, the agent updates the estimated value function of S_t based on the reward received and the estimated value of the current state. $V(S_t)$ thereby is the old estimate, and $R_{t+1} + \gamma V(S_{t+1})$ is the so-called target. This is expressed in the update rule:

$$V(S_t) = V(S_t) + \alpha[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)] \tag{4.13}$$

where α is the so-called learning rate, which specifies the rate at which information is transferred to the value function estimate [46]. The difference between target and estimate $[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$ is referred to as the TD-error. The key feature of TD-Learning is that it updates estimates based on other estimates (bootstrapping), thus allowing learning before the final outcome is known, contrary to Monte Carlo methods. This leads to faster convergence, making TD-Learning an attractive method for solving

RL problems where complete knowledge of the environment is not available, or where the concept of finite episodes does not apply. What is more, TD-Learning constitutes a fundamental theoretical advancement which significantly facilitated the development of DRL methods.

Q-Learning

Before going into DRL, one last idea will be explained due to its fundamental importance in RL: Q-Learning. Q-Learning attempts to learn the action-value function $Q(s, a)$ from experience collected during training without requiring a complete policy iteration at each step of the algorithm [60]. In contrast to Value Iteration, which requires full knowledge of the environment dynamics, Q-Learning works by updating the action-value function (Q-value) based on the experience the agent collects. Q-Learning doesn't require model dynamics because it does not only consider the state, but also the action taken to reach a given state. Over time, this leads to an understanding of which action to take in each state. Q-Learning therefore is another early yet important implementation of model-free RL.

The Q-values are updated iteratively via the Bellman equation, which incorporates the maximum expected future rewards for the next state, as shown below:

$$Q(s, a) = Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)] \tag{4.14}$$

where α is the learning rate, s' is the new state after taking action a, and γ is the discount factor. The new Q-value $Q(s, a)$ is made up of the previous estimate of $Q(s, a)$ and the approximation error compared to the target - this error is also referred to as TD-error due to its resemblance with TD-Learning.

4.3 Deep Reinforcement Learning

Deep Reinforcement Learning (DRL) leverages technological advancements in the field of deep learning to approximate value functions and policies with deep neural networks [46]. As shown in Chapter 2, DRL led to algorithms with unparalleled performance, even when compared with the human benchmark. This section will dive into neural networks, the theory of DRL, and explain the state-of-the-art algorithms TD3 and PPO, which are predominantly used in this study. For further readings on

other modern DRL algorithms, consider [46, 56, 61–63].

4.3.1 Deep Neural Networks

As the name suggests, DRL involves the use of Deep Neural Networks (DNNs). Neural networks are computational models inspired by biological neural networks, and are made up of artificial neurons or nodes. Each node performs simple computations: it aggregates inputs a_i that are scaled with weights w_i, and a bias term b, and then passes this sum z through an activation function $g(z)$, as illustrated in Figure 4.3.1. Activation functions can be linear, piecewise linear (e.g., ReLU), or non-linear (e.g., tanh or sigmoid) [64]. Nodes are organized into layers, where each layer can have a different number of nodes. The layers between the input and output layers are referred to as hidden layers. A neural network is considered ”deep” if it contains two or more hidden layers.

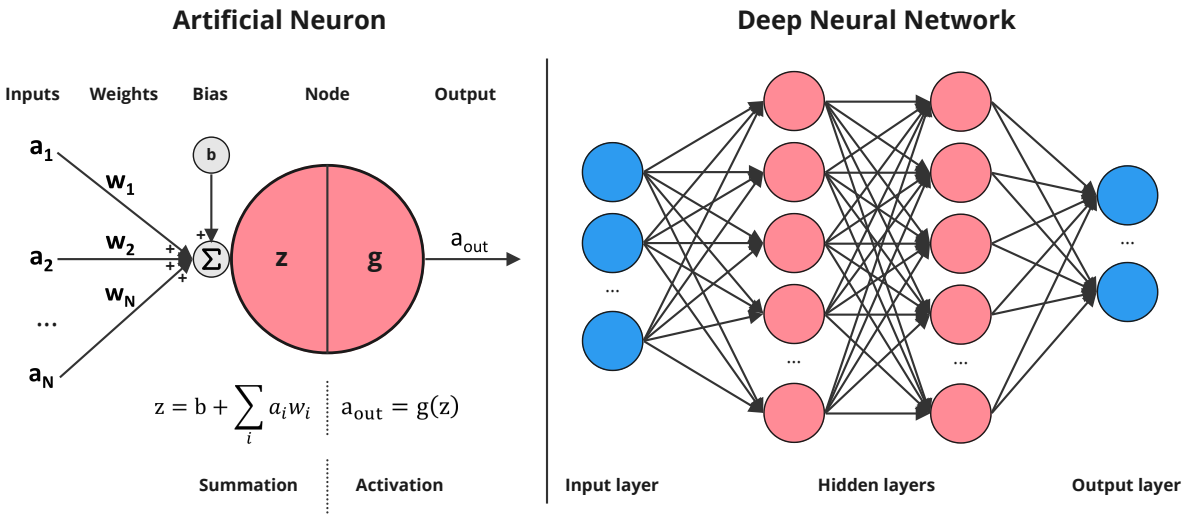


Figure 4.3.1: Artificial Neuron and DNN

Figure 4.3.1 depicts the structure of a Multi Layer Perceptron (MLP), a type of neural network where all nodes in each layer are fully connected to the nodes in the neighbouring layers. Initially, all weights and biases are initialized with small random values. The network then updates its weights and biases, trying to minimize a loss function that calculates the error between prediction and ground truth [65]. In an MLP, the different layers do not perform explicitly different computational tasks - the network as a whole solves the given problem without having to specify how it goes

about solving it. This makes the MLP a versatile architecture suitable for a wide range of numerical problems [64].

A Convolutional Neural Network (CNN), on the other hand, also performs well on grid-like data such as images. CNNs excel in pattern recognition and image classification tasks due to their unique layer architectures. In a CNN, the recognition layers, called convolutional layers, typically handle specific aspects of a larger problem-solving task. For example, earlier layers might detect edges and simple features, whereas later layers recognize more complex features within an image [66]. The convolutional layers are usually followed by one or more fully-connected layers with MLP architecture, that perform the final classification or regression task. CNNs are not used in this study due to the numeric nature of the EV charging problem, but can find applications in image-based RL problems [49].

4.3.2 Deep Q-Learning

Deep Q-Learning encompasses all methods in which the action-value function $Q(s, a)$ is approximated via a DNN. Arguably, the most significant contribution in the field was made by DeepMind with their CNN-based algorithm Deep Q-Network (DQN), in which a DRL agent was trained to play Atari 2600 games at unprecedented skill levels [49]. Deep Q-Learning and DQN are often used interchangeably because DeepMind's implementation set a new state-of-the-art regarding the implementation of Q-Learning with DNNs.

Deep Q-Learning compares the current estimation of $Q(s, a)$ with a target and tries to minimize the difference. The target thereby is the Bellman backup for the action-value function: $R(s, a) + \gamma \cdot \hat{Q}(s', a')$, where $\hat{Q}(s, a)$ is the target Q-value. The DNN forms the heart of the algorithm, as it is responsible for predicting the best action at every time step. In fact, two DNNs are used: one to compute the Q-value $Q(s, a)$ at every iteration, and one to compute the Q-value target $\hat{Q}(s, a)$ every C iterations [49]. The weights of the Q-network θ_i are updated each step, and the weights of the target Q-network θ_i^- are updated every C steps. The decoupling of Q and target Q-value estimation increases stability and avoids oscillations or divergence by a great extent [49].

What further distinguishes DQN from Q-Learning is its biologically inspired use of memory. Using the memory - also referred to as experience replay - the agent can store past actions and replay them, allowing for a more efficient training process. Learning

from past transitions occurs in batches that are sampled from the experience replay. Algorithms such as DQN that learn from stored historical data are known as *off-policy* algorithms. Their counterparts, *on-policy* algorithms, on the other hand, only learn from live updates directly collected from the environment and discard this experience afterwards, making on-policy algorithms generally less sample efficient. Besides the use of Neural Networks and experience replay, the algorithm is very similar to tabular Q-Learning. Figure 4.3.2 visualizes the entire learning process, and each step is further explained after the figure.

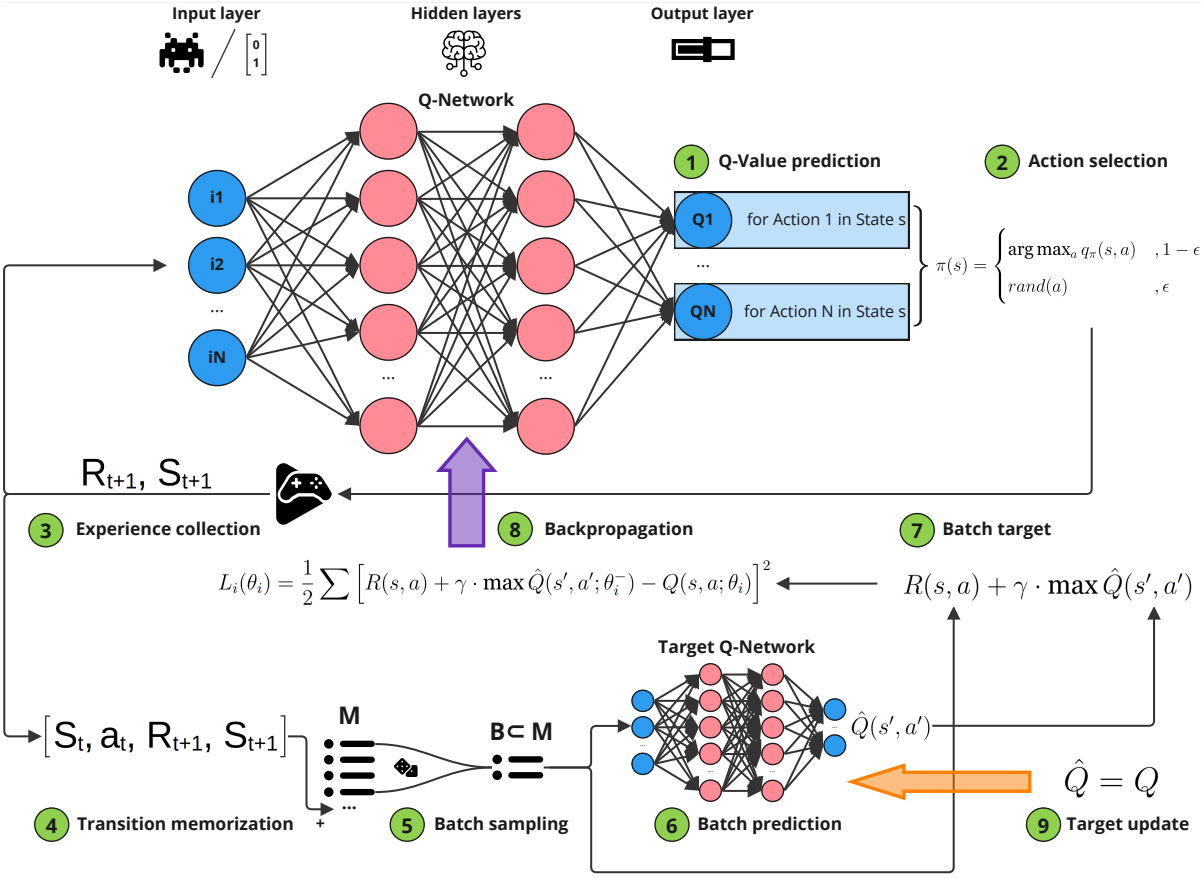


Figure 4.3.2: Schematic overview of the learning process in a DQN

1. Q-value prediction: Given the current state s, the Q-value $Q(s,a)$ is predicted for each possible action using the Q-network.
2. Action selection: Following an ϵ-greedy policy, the agent either chooses the action yielding the highest Q-value, or a random action to explore the environment.
3. Experience collection: The chosen action is experienced in the environment, yielding a reward signal and the next state, making up the loop of the MDP.

4. Transition memorization: The transition $S_t, a_t, R_{t+1}, S_{t+1}$ is appended to M.
5. Batch sampling: A subset $B \subset M$ is randomly sampled from M.
6. Batch prediction: The target Q-values $\hat{Q}(s, a)$ are predicted for the entire batch using the target Q-network.
7. Batch experience: Given the experienced reward from memory and the next best $\hat{Q}$-value, the target is calculated: $R(s, a) + \gamma \cdot \max \hat{Q}(s', a')$.
8. Backpropagation: The Mean Squared Error (MSE) loss function L is computed for the difference between prediction and target. Note that the loss function is a function with many variables and parameters: the weights θ_i in the neural network (variables) and the weights of an earlier iteration θ_i^- from the target Q-network (parameters):

$$L_i(\theta_i) = \frac{1}{2} \sum \Big[R(s, a) + \gamma \cdot \max \hat{Q}(s', a'; \theta_i^-) - Q(s, a; \theta_i) \Big]^2 \tag{4.15}$$

$L_i(\theta_i)$ is differentiated with respect to θ_i (chain rule), yielding the gradient ∇_{θ_i}:

$$\nabla_{\theta_i} = \big(R(s, a) + \gamma \cdot \max Q(s', a'; \theta_i^-) - Q(s, a; \theta_i) \big) \nabla Q(s, a; \theta_i) \tag{4.16}$$

The weights in the neural network are then adjusted slightly towards the direction that minimizes the loss function - this is known as gradient descent.

9. Target update: Every C steps, reset the target Q-network: $\hat{Q} = Q$

4.3.3 Actor-Critic Algorithms

While Deep Q-Learning already achieves considerable improvements compared to its tabular counterpart, it still has practical limitations because it can only deal with discrete action spaces. This is because of the way the action is chosen: the *argmax* function can be trivially computed for the discrete case, but becomes challenging to evaluate in the continuous space. Therefore, to transition to the continuous case, a second neural network is added that learns which actions to take based on the given state, replacing step 2 in Figure 4.3.2. This so-called policy-network, also known as an actor, aims to maximize the reward received. During training, gradient ascent is conducted: the weights of the network ϕ are updated such that the return function $J(\phi)$ increases in value. The updates are conducted according to the Deep Deterministic

Policy Gradient (DDPG) algorithm [67, 68].

$$\nabla_\phi J(\phi) = \frac{1}{N}\left[\sum_i^N \nabla_\phi \pi_\phi(s) \nabla_a Q^\pi(s,a)|_{s=s_i, a=\pi(s_i)}\right] \tag{4.17}$$

As might have been anticipated from Eq. 4.17, the gradient depends on the slope of the policy function, as well as the slope of the Q-value function. If said value function originates from a neural network that outputs Q-values (the so-called critic network), an actor-critic architecture is formed. This is shown in Figure 4.3.3. Popular implementations of this architecture are DDPG, Soft Actor Critic (SAC), TD3, and PPO [61–63, 68].

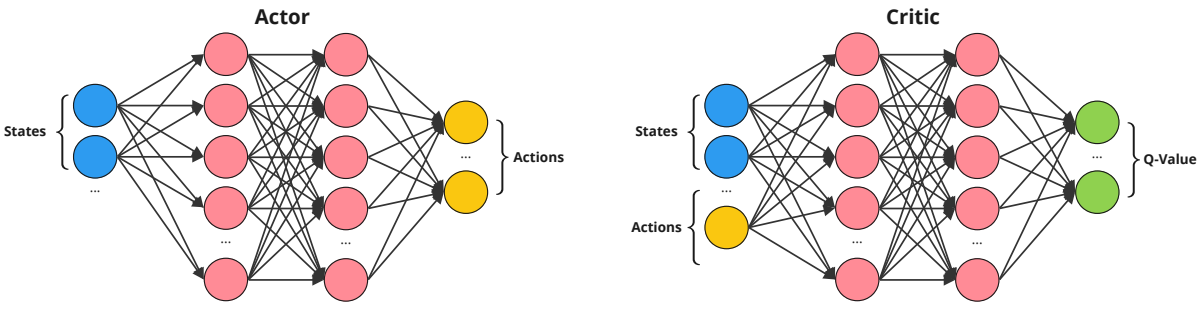


Figure 4.3.3: Actor-Critic Architecture

Since the critic network is a neural network with weights θ_i, the update rule for the actor network's weights ϕ_i is as follows according to Eq. 4.17:

$$\nabla_\phi J(\phi) = \frac{1}{N}\left[\sum_i^N \nabla_a Q(s,a;\theta)|_{s=s_i, a=\pi(s_i;\phi)} \nabla_\phi \pi(s;\phi)|_{s=s_i}\right] \tag{4.18}$$

4.3.4 Twin Delayed Deep Deterministic Policy Gradient

Twin Delayed Deep Deterministic Policy Gradient (TD3) implements an actor-critic infrastructure and addresses issues such as instability and overestimation bias, making it one of the state-of-the-art DRL algorithms [68]. The algorithm consists of a total of six DNNs: one actor, one actor-target, two critic, and two critic-target networks. The two critic-target networks both estimate the Q-value target $\hat{Q}$, but only the smaller value is used to backpropagate error into the Q-network to avoid overestimation. Additionally, a delay is added when backpropagating into the actor network to

minimize the risk of instability or oscillations [68]. Figure 4.3.4 summarizes the entire learning process, with further explanations on each step after the figure.

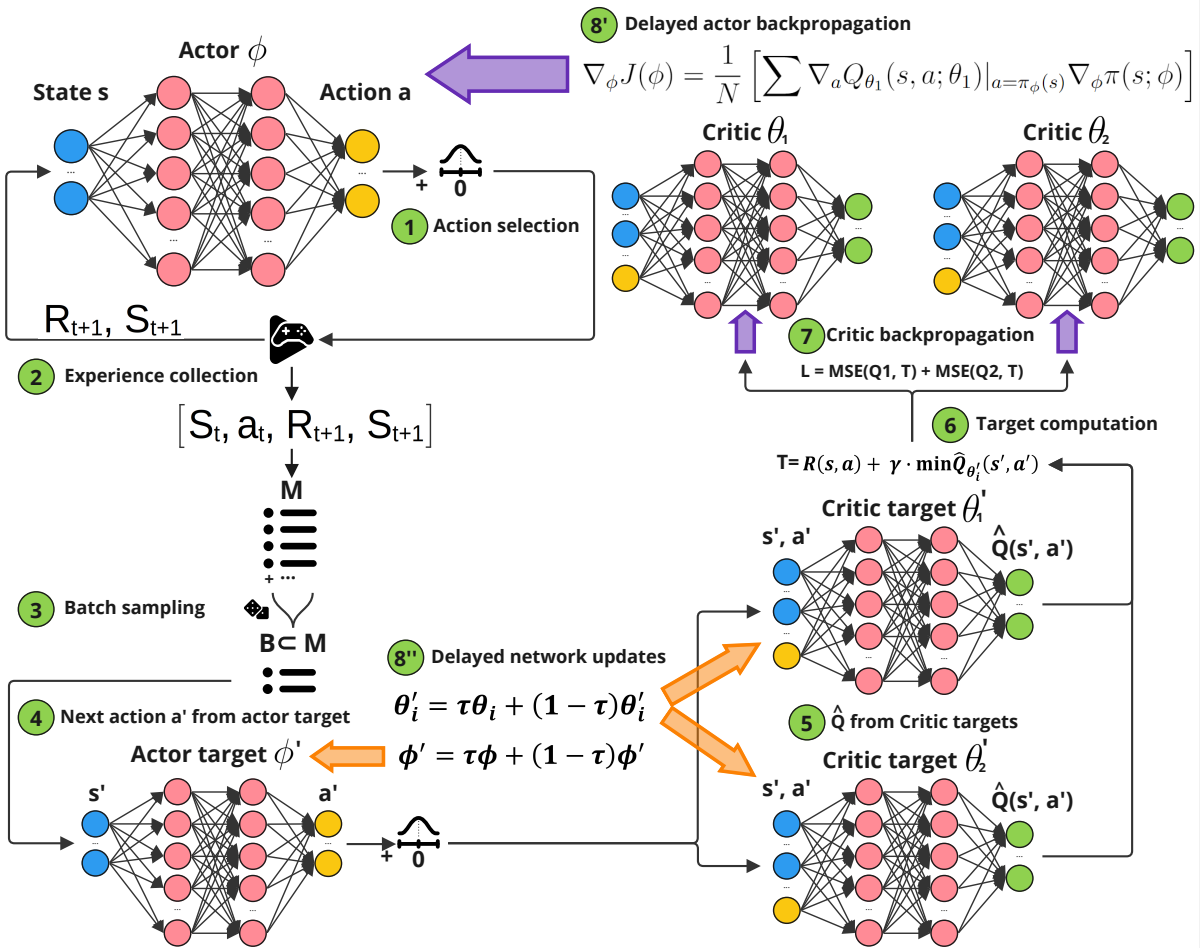


Figure 4.3.4: Schematic overview of the learning process in TD3

1. Action Selection: An action is selected from the actor network ϕ. Gaussian noise is added to the action to ensure sufficient exploration by the agent and to avoid getting stuck in local minima.

2. Experience collection: The action is sent to the environment, yielding the next state and the reward. This makes up the MDP loop. The tuple $(S_t, a_t, R_{t+1}, S_{t+1})$ is stored in M.

3. Batch sampling: A subset $B \subset M$ is randomly sampled from M.

4. Next action a' from actor-target: The actor-target takes in the next states $S_{t+1} = s'$ from the batch and predicts the next corresponding actions $A_{t+1} = a'$.

5. $\hat{Q}$ from critic-targets: The critic-targets predict the target Q-value $\hat{Q}(s', a')$.

6. Target computation: The target T is computed using the reward from the batch and the minimum target Q-value to avoid overestimation bias.

7. Critic backpropagation: The loss function consists of the sum of both MSE values. Similar to Eq. 4.16, the weights are updated via gradient descent, such that $\theta_i = \arg\min_{\theta_i} L(\theta_i)$, where θ_i are the weights of networks $i = 1$ or $i = 2$.

8. Delayed actor backpropagation and network updates: Every d steps, the actor is updated via the DDPG algorithm, similar to Eq. 4.18. On this iteration, the target networks are updated as well. τ thereby represents the rate at which information from the networks is transferred to their respective target counterparts.

4.3.5 Proximal Policy Optimization

PPO aims to address the complexity, sensitivity, and brittleness of tuning that can occur with other policy gradient methods [61]. Unlike DQN and TD3, PPO is an on-policy algorithm: it only learns from batches of immediate experience collected and discards each batch after the learning update. Since PPO does not have a replay buffer to look back to, it is vulnerable to large policy updates that send the agent into the wrong direction, potentially causing convergence in undesirable local minima [61]. To prevent this, PPO limits the update size by clipping its loss function, therefore creating a region of trust in which an update can be safely performed. Due to its performance, ease of use, and ease of implementation, PPO is recognized as a state-of-the art DRL algorithm, and research efforts are being put into further improving the framework [61, 69].

PPO is an actor-critic implementation with one actor and one critic network. However, the structure differs slightly from the actor-critic implementations shown above, where the actor network takes in states and outputs corresponding actions. In PPO, the actor outputs the policy distribution instead: for each action a_i, the network outputs a mean value μ_{a_i} and a standard deviation σ_{a_i}. To actually select an action, one has to sample from this distribution. During training, the shape of this distribution is changed such that the loss function is minimized. Additionally, the critic network in PPO estimates state-values $V(s)$ instead of state-action values $Q(s, a)$. Note how both actor and critic now take in states in the input layer and only differ in the output layer. Due to their similarity, the two networks can share weights and biases, which reduces complexity and increases computational speed. In practice, however, it has been found that a

shared architecture performs worse due to the conflicting goals of predicting actions and values with the same parameters, which is why separate actor and critic networks are more commonly implemented [69]. The entire learning process of PPO is shown in Figure 4.3.5. Further explanations on each step are provided after the figure.

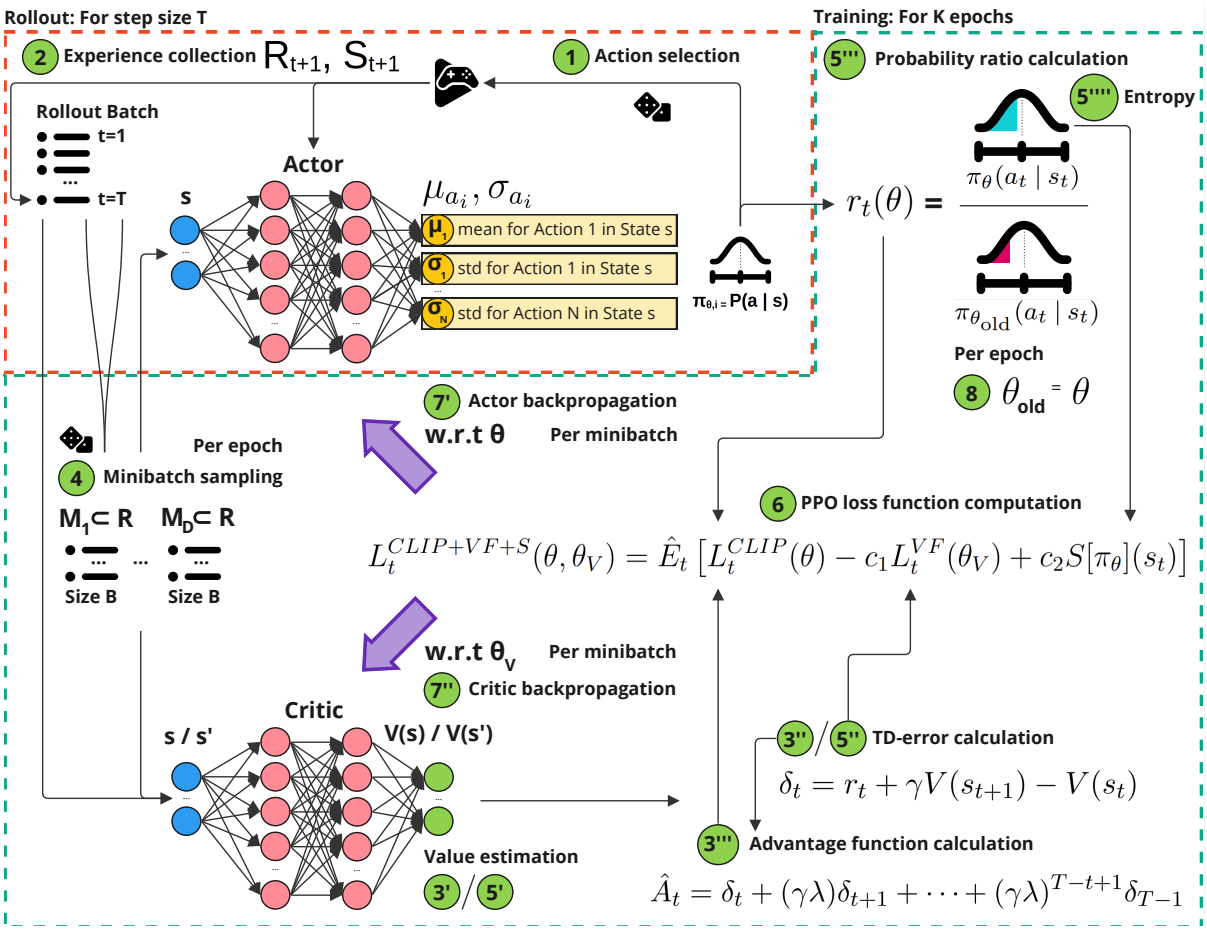


Figure 4.3.5: Schematic overview of the learning process in PPO

The learning process in PPO is divided into two phases: roll-out and training. During the roll-out phase, the current policy is deployed in the environment. For T time steps, actions are sampled from the distribution, yielding $a_1, ..., a_T$. The transitions $S_t, A_t, R_{t+1}, S_{t+1}$ are thereby stored in the so-called Rollout Batch of size T. Once the roll-out phase is completed, the learning phase starts. In a first pass, for each time step t in the rollout batch, the state-value and TD-error of the taken action a_t are computed. Naturally, the TD-error can be used for the critic loss function, such that the critic network learns to reduce the MSE between value prediction and value target.

$$L_V(\theta_V) = \frac{1}{2} \sum \left[R_{t+1} + \gamma \cdot V(s_{t+1}; \theta_V) - V(s_t; \theta_V) \right]^2 \tag{4.19}$$

However, the TD-error is also used to compute the advantage function $\hat{A}$ once all TD-errors until $t = T$ have been evaluated. The advantage function $\hat{A}_t$ is defined as the discounted sum of TD-errors δ_t for $t = t, t+1, ..., T$:

$$\hat{A}_t = \sum_{l=0}^{T} (\gamma\lambda)^l \delta_{l+t} \tag{4.20}$$

where λ is used to strike a balance between bias and variance during estimation [70]. Note the recursive nature of the function: $\hat{A}_{T-1} = \gamma\lambda\hat{A}_T$, etc. The advantage function $\hat{A}_t$ uses the TD-errors δ_i to compare the most recent value to the previous value estimation. A positive advantage function therefore represents an improvement, indicating that a better action was taken for state s_t than the previous estimate, and vice versa. This metric guides the agent in the right direction, as it always incentivizes beating the previous estimate. The advantage function is calculated for each timestep in the rollout batch before any network is updated. This ensures that the advantage function represents the performance of the policy that was used during the rollout phase.

Once this first pass is completed and the advantage function has been evaluated for all entries in the rollout batch, network updates can be conducted. Updates occur in epochs - a machine learning term that indicates an iteration over the entire dataset. In this case, one epoch corresponds to an iteration over the rollout batch. In each epoch, the rollout batch is split into D randomly sampled minibatches of size B. In this process, the temporal coherency of the trajectory is lost to avoid correlation between consecutive states. Each minibatch is then processed separately, and a network update is performed once per minibatch. After all minibatches have been processed, one epoch is completed.

To conduct an update, the prerequisites of the loss function have to be calculated; both for the critic and actor network. As mentioned above, the TD-error is used for the loss function of the critic network, as shown in 4.19. Note that the state-values $V(s; \theta_V)$ are now changing with each update of θ_V - the TD-error can therefore change slightly after each minibatch.

The critic loss function is made up of two main prerequisites: the advantage function and the policy change. The policy change involves the probability of action a_t. It is used to compare the ratio of probabilities $r(\theta)$ which provides a metric on how much

the policy changed since the last actor update. This ratio is later clipped in the loss function to avoid instability due to excessively large policy updates. The old policy π_{old} thereby is a snapshot of the old actor network.

$$r(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \tag{4.21}$$

With the prerequisites fulfilled, the actor loss function can be calculated. The actor loss function in PPO is also known as a surrogate objective function. Its main part consists of the expression $r_t(\theta)\hat{A}_t$. This expression is positive if the last action beat the previous value estimation (positive advantage function), and it is negative, if the last action performed worse than the previous value estimation (negative advantage function). The magnitude of the expression is scaled by the probability ratio $r(\theta)$ to incorporate the information by how much the policy changed since the last update. Significant stability increases are further achieved by clipping $r_t(\theta)$ between $1-\epsilon$ and $1+\epsilon$ to limit the maximum policy change [61].

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t \left[min(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t) \right] \tag{4.22}$$

Finally, an entropy term is added to the clipped surrogate objective to introduce exploration noise [61].

$$S[\pi_\theta](s_t) = \hat{\mathbb{E}}_t \left[\pi(a_t|s_t;\theta) \cdot \log \pi(a_t|s_t;\theta) \right] \tag{4.23}$$

According to [61], the total PPO loss function is thus expressed as follows:

$$L_t^{CLIP+V+S}(\theta, \theta_V) = \hat{E}_t \left[L_t^{CLIP}(\theta) - c_1 L_t^V(\theta_V) + c_2 S[\pi_\theta](s_t) \right] \tag{4.24}$$

where c_1 and c_2 are scaling parameters. Slight changes have been made to the notation to account for the fact that actor parameters θ and critic parameters θ_V can be different. The loss function is computed for each entry in the minibatch. Before backpropagation, an average is computed over all entries. When backpropagating the loss function with respect to θ, the following update is obtained:

$$\theta = \theta - \alpha_\theta \nabla_\theta \left[L^{CLIP}(\theta) + c_2 S[\pi_\theta](s_t) \right] \tag{4.25}$$

where α_θ is the learning rate of the actor network. When backpropagating with respect to θ_V, on the other hand, following update is obtained:

$$\theta_V = \theta_V - \alpha_{\theta_V} \nabla_{\theta_V} \left[-c_1 L^V(\theta_V) \right] \tag{4.26}$$

After each epoch, the snapshot of the old actor network θ_{old} is set to the state of the actor network θ.

Finally, the entire training process in PPO is explained step by step below.

Rollout phase: for $t = 1, ..., T$ under policy π_θ

1. Action selection: An action is sampled from the policy distribution π_θ of the actor network.

2. Experience collection: the action is sent to the environment, yielding the next state and the reward. This makes up the MDP loop. The trajectory is stored in the rollout batch R.

Training phase

3. Advantage function calculation

   (a) Value estimation: The state s and next state s' in the trajectory are sent through the critic network, yielding $V(s)$ and $V(s')$. This is repeated for the entire rollout batch.

   (b) TD-error calculation: The trajectory's reward, $V(s')$ and $V(s)$ are used to calculate the TD-error $\delta_t \; \forall t$.

   (c) Advantage function calculation: After all δ_t have been calculated, $\hat{A}_t$ can be computed $\forall t$. Note that the advantage function is held constant for all epochs of the current training phase.

4. Minibatch sampling: For each epoch, random minibatches $M_i \in R$ with minibatch size B are sampled. The total number of minibatches thereby is the ratio of step size and minibatch size: $D = \frac{T}{B}$.

5. Loss function prerequisites:
   (a) Value estimation: Once more, the value function is estimated for s and s' for each transition in the minibatch.
   (b) TD-error calculation: The TD-error is calculated once again. This time, however, it is used for the critic network's loss function.
   (c) Probability ratio calculation: Via the actor network θ and its predecessor θ_{old}, the probabilities $\pi(a|s)$ are calculated for each state. The fraction $r(\theta)$ is computed for each state in the minibatch.
   (d) Entropy bonus: The entropy bonus is added for each state in the minibatch to introduce exploratory action noise.
6. PPO loss function calculation: With all prerequisites calculated, the PPO loss function can be computed. It consists of the clipped surrogate loss, an entropy bonus, and the value loss function. The loss function is computed for all entries of the minibatch. Then, either an average or a sum is used for backpropagation.
7. Backpropagation:
   (a) Actor backpropagation: The PPO loss function is differentiated with respect to θ, yielding the update shown in Eq. 4.25. This is done once per minibatch.
   (b) Critic backpropagation: The PPO loss function is differentiated with respect to θ_V, yielding the update shown in Eq. 4.26. This is done once per minibatch.
8. Old actor network update: After an epoch has been completed, θ_{old} is set to θ.

Having covered the theoretical fundamentals, as well as the advanced DRL algorithms that are used in this study, the next chapter will explain how the EV charging problem was modelled, and how the algorithms are used to optimize EV charging.

Chapter 5

Methodology

This chapter explains the methodological approach of this study. Three sections split the chapter in its sub-components: Use-case design, quantitative analysis, and economic impact assessment, as shown below. In each section, a detailed account will be given on the reasoning and approach, as well as the contribution towards answering the research questions.

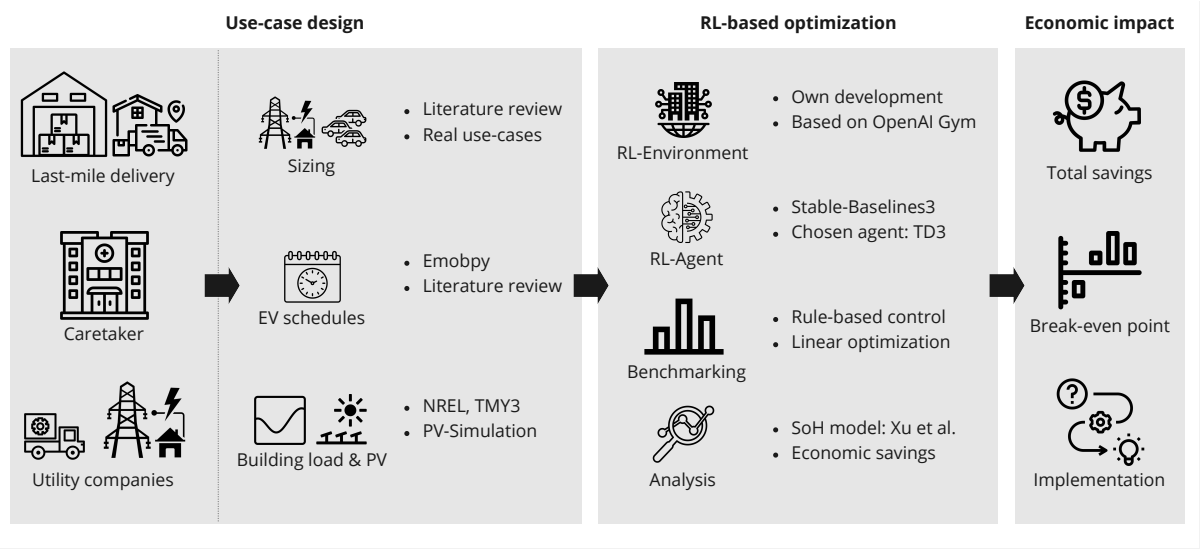


Figure 5.0.1: Methodology overview

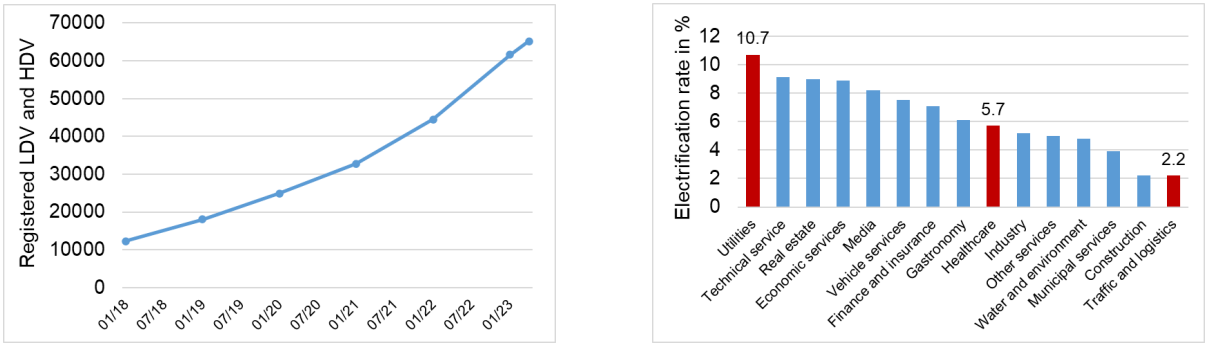
5.1 Use-case design

Commercial use-cases are chosen over residential applications for three reasons. First, a research gap has been identified, when it comes to optimizing commercial fleet charging with reinforcement learning, as shown in Chapter 2. Second, the demand

for an optimized EV charging process is higher in commercial applications due to high sensitivities for down-times and operational costs. Finally, the optimization of a fleet can potentially have a higher impact in terms of economics and sustainability, because the number of vehicles optimized per customer is significantly higher in commercial applications than in residential use-cases. With registration numbers for commercial light- and heavy-duty-vehicles growing steadily, as shown in Figure 5.1.1, the analysis of commercial fleets becomes even more relevant.

Different commercial applications can vary significantly in terms of fleet size, vehicle type, building load and grid connection capacity. Therefore, the suitability for RL-based charging optimization might vary with different applications. To take this into account, three commercial use-cases are modelled to allow for sensitivity in certain input parameters. The use-cases are meant to be representative for the three different commercial activities, so that general learnings can be drawn from this study. Wherever possible, assumptions are backed up by evidence or scientific literature; in some cases, conservative assumptions are considered. The inputs are integrated into the model in such a way that they are easily interchangeable, thus making the model applicable to a wide range of use-cases.

In this study, the use-cases of last-mile delivery, caretakers, and utility maintenance vehicles were considered. These three cases were chosen because they differ considerably in their schedules, vehicle types, and challenges regarding electrification. When considering German mobility statistics, the differences become evident, as shown in Figure 5.1.1 [11].



(a) Registered LDVs and HDVs in Germany (b) Electrification by commercial sector

Figure 5.1.1: EV uptake in duty vehicles and electrification by commercial sector [11]

This section will first describe the general methodology of compiling the use-cases, including schedule and load profile generation, company sizing, and data sources.

Then, the use-cases will be explained, highlighting their distinctive characteristics and input parameters, as well as challenges that the sectors face in terms of fleet electrification.

5.1.1 Design approach

Vehicle schedules

Since EV schedules are essential when it comes to EV charging optimization, their generation is well-discussed and documented in scientific literature. Several frameworks exist to accomplish this; emopby is amongst the most sophisticated and well-accepted [71]. Emobpy does not only generate vehicle schedules based on mobility statistics, but also performs detailed calculations on driving consumption. The framework has been applied to Germany via openly available mobility statistics and the dataset is made public [72]. The dataset depicts the mobility patterns of the general German population with 119 representative vehicles over a time horizon of one year and a resolution of 15 minutes. It includes the data, arrival and departure times, distance travelled and consumption in kWh, and other parameters, as shown in Table 5.1.1.

Date	ID	Location	Distance_km	Consumption_kWh	ChargingStation	PowerRating_kW
03/01/2020 17:30	0	shopping	0	0	none	0
03/01/2020 17:45	0	shopping	0	0	none	0
03/01/2020 18:00	0	shopping	0	0	none	0
03/01/2020 18:15	0	driving	3.667	0.634	none	0
03/01/2020 18:30	0	driving	3.667	0.634	none	0
03/01/2020 18:45	0	driving	3.667	0.634	none	0
03/01/2020 19:00	0	home	0	0	home	3.7
03/01/2020 19:15	0	home	0	0	home	3.7
03/01/2020 19:30	0	home	0	0	home	3.7

Table 5.1.1: Emobpy data set for the German case [72]

With its thorough documentation, online community and already existing application to the German case, emobpy seemed like a suitable framework for the schedule generation for the commercial use-cases. However, emobpy's logic is designed around the modelling of residential schedules, e.g. commuting to work, or to shopping, as shown in Table 5.1.1. The attempt to use it for the generation of commercial schedules proved to be complex, because the trip characteristics of commercial vehicles vary significantly from residential vehicles. While a delivery vehicle would depart the depot and return 150 km and 10 hours later, a personal vehicle would make several stops to shopping, leisure, or work and return home, only travelling a small distance between

each stop and remaining at each stop for some time. In order to model a delivery schedule in emobpy, one would have to know the distance and idle time of delivery for every single parcel, thus making it impractical to use.

Therefore, a dedicated schedule generation algorithm was developed in Python that incorporates the use-case's mobility characteristics and statistics on vehicle consumption, and outputs a one-year vehicle schedule. Every trip, a random sample is drawn from a uniform distribution that follows the mobility and vehicle consumption statistics. Regarding mobility, the mean arrival and departure time, as well as the mean distance are required, along with their standard deviations. These parameters were set according to the literature review and conservative assumptions. Regarding vehicle consumption, the mean value was set to the consumption of the chosen vehicle model stated by the manufacturer. The standard deviation, minimum and maximum value, however, were set according to the emobpy dataset for Germany to incorporate representative fluctuations due to temperature, wind speed, and other factors.

Sizing of fleet and building

The fleet size, surface area of the building and PV panels, and the grid connection capacity were mostly taken from real-world companies that are electrifying their fleet, and from conservative assumptions in some cases. When real-world companies were considered, further research was done to ensure that they are representative for the industry. The information platform *electrive.net* was used to gather the relevant information [73]. The platform aggregates press releases of companies across Germany, making it easy to look up information on recent commercial cases of fleet electrification. Via information on Electrive, and further research on the respective companies, it was possible to find information on the sizing of the fleet, building, and PV panels. Regarding the grid connection, it was not possible to find accurate information. The grid connection was therefore assumed such that the simultaneous charging of more than 50% of the electric vehicles would result in an overloading.

Building load

The building load and its time series were sourced from the Open Energy Data Initiative of the National Renewable Energy Laboratory [74]. The data is available either in form of end-use load profiles of individual buildings, or in form of aggregated profiles. For commercial buildings, the aggregated profiles are compiled to represent reference

buildings with a certain size, such as a warehouse or a small office [75]. The data is available by state, and the state of Washington was chosen due to its resemblance with the German climate [76]. Depending on the use-case's building and its size, the corresponding reference profile was chosen and scaled linearly with surface area.

PV generation profile

The capacity factors of PV in Germany were sourced from the MERRA-2 dataset [77]. The used dataset consists of hourly capacity factors for Germany in 2019. The capacity factors were then multiplied with the install capacity at the use-case site to yield generation profiles. The profiles of each company therefore have the same shape and different magnitudes. If available, more granular data could be used to model the generation at a specific company's location.

5.1.2 Use-cases

Last-mile delivery

Last-mile delivery is the transportation of goods from the distribution centre to the end customer. It involves comparatively short distances and time frames - e.g. same-day delivery. Last-mile delivery usually finds itself at the last step of the logistics chain of e-commerce business models, but can also be applied to other use-cases, such as delivering food or ingredients to supermarkets or restaurants. Example companies could be conventional parcel companies such as DHL, or sustainably orientated companies such as Fairsenden, which already advertise with low-CO2 and electrified delivery services [78].

Trips in last-mile delivery are typically characterized by frequent journeys with short distances, of around 80-165 km [79]. The vehicles operate during usual business hours and return to centralized depots at the end of the shift, where they remain until their next deployment. Usually, operations also take place on Saturdays, however with reduced activity. Table 5.1.2 shows the parameters for the last-mile delivery vehicle schedules.

Parameter	Mean	Standard deviation
Distance Weekday	150 km	25 km
Distance Saturday	75 km	25 km
Departure Weekday	07:00	1.0 h
Arrival Weekday	19:00	1.0 h
Departure Saturday	09:00	1.5 h
Arrival Saturday	17:00	1.5 h

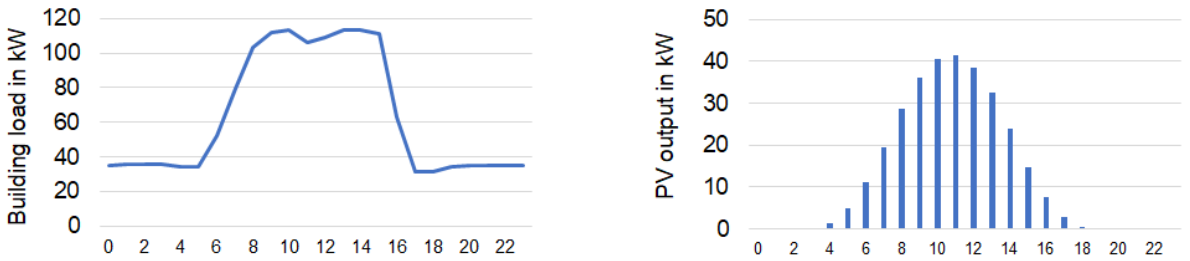
Table 5.1.2: Last-mile delivery schedule parameters

Large light-duty vehicles are required for the transport of goods and parcels to ensure enough storage room and battery capacity. In this case, the Mercedes Benz e-Vito Tourer was chosen for the case-study [80]. Its specifications are shown in Table 5.1.3. The consumption mean is taken from the manufacturer, and the standard deviation, minimum, and maximum values are computed from the emobpy dataset, as mentioned above [72, 80].

e-Vito	Value	Unit
Range	314	km
Battery capacity	60	kWh
Engine power	85	kW
Charging power	11/22	kW AC
Consumption mean	0.213	kWh/km
Consumption std. deviation	0.155	kWh/km
Consumption min	0.193	kWh/km
Consumption max	0.453	kWh/km

Table 5.1.3: Last-mile delivery vehicle specifications

The warehouse was chosen as the commercial reference building [75]. Its size was scaled to $10000\,m^2$ or $1\,ha$, which is within the usual range for a medium logistics centre [81]. Its minimum, maximum, and average building load amount to 14, 190, and 63 kW, respectively. Similarly, a fleet size of 50 vehicles was chosen to form a medium-sized vehicle fleet [73, 79]. To charge the vehicles, charging points of 11 kW were chosen. PV was assumed to cover 5% of the rooftop area with a power density of $175\,\frac{W}{m^2}$, equalling an install capacity of $87.5\ kW$. The maximum and average power output are 63.4 kW, and 11.1 kW, respectively. Given the fleet size, building load and PV, a load of around 500 kW would occur if half the vehicles were to charge simultaneously during a lack of PV generation. The grid connection capacity was thus set at 500 kW. Using the TMY-3 and MERRA-2 datasets, the profiles were created, as shown in Figure 5.1.2.



(a) Building load (b) PV power output

Figure 5.1.2: Building load and PV output for last-mile delivery

Last-mile delivery companies face different challenges regarding electrification, depending on the company size. Large players see fleet electrification as a mean to gain competitive advantage through marketing campaigns and sustainable value propositions. As long as their operation reliability is not impacted, they are willing to commit to high capital expenditures and the associated economic risk [79, 82]. Medium-sized companies might have the means necessary to purchase an electric fleet, but they are more risk-averse and need to ensure profitability. A possible measure therefore consists of subcontracting the fleet operation to a third-party company specialized in vehicle handling; a practice also commonly seen in the segment of small delivery companies [82–84]. With decreasing company size, vehicles are also used more versatilely, making fossil-fuelled alternatives more attractive. Generally, tax incentives or other benefits might be required to ensure economic feasibility [79].

Caretaker

Caretakers provide healthcare services at the patient's home, typically serving a certain municipal district or the local rural communities. Thus, their schedules are characterized by frequent and short-distance journeys that do not stray far from the depot location [85]. Due to the constant demand for nursing, caretakers also operate during the weekend. Unscheduled trips are possible due to urgent calls and emergencies, which is why a 2% probability was incorporated that a trip occurs during the night. In the caretaker business, it is common that vehicles return to their depot during the day, for example due to break times or for refuelling purposes. This characteristic was adopted from the *SMobilityCOM* project, which analysed the feasibility of EV fleets in the caretaker industry via real use-cases in Germany [85]. According to the project, a break during the day might even be necessary to refuel the short-ranged vehicles for the afternoon shift. Table 5.1.4 shows the parameters for the

caretaker vehicle schedules.

Parameter	Mean	Standard deviation
Distance Weekday	30 km	10 km
Distance Weekend	15 km	15 km
Departure Weekday	06:00	1.0 h
Pause begin	12:00	0.25 h
Pause end	13:30	0.25 h
Arrival Weekday	19:00	1.0 h
Departure Weekend	07:00	1.5 h
Pause begin	12:00	0.25 h
Pause end	13:00	0.25 h
Arrival Weekend	15:00	1.5 h
Probability of emergency	2%	-

Table 5.1.4: Caretaker schedule parameters

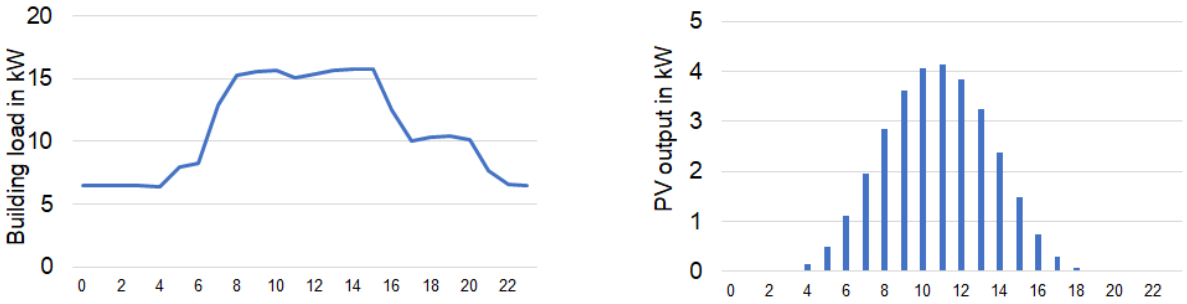
Since caretaker companies do not transport cargo or patients, their fleet can be made up of relatively small vehicles. Considering recent cases of fleet electrification in the caretaker sector, the models *VW e-UP!* and *Smart EQ fortwo* seem particularly popular [86]. For this case study, the electric Smart was chosen [87]. Its specifications are shown in Table 5.1.5.

Smart EQ fortwo	Value	Unit
Range	90	km
Battery capacity	16.7	kWh
Engine power	60	kW
Charging power	4.6	kW AC
Consumption mean	0.17	kWh/km
Consumption std. deviation	0.155	kWh/km
Consumption min	0.193	kWh/km
Consumption max	0.453	kWh/km

Table 5.1.5: Caretaker vehicle specifications

For the caretaker company site, the small office was chosen as the commercial reference building. Taking the caretaker company from [86] as an example, the surface area was scaled to $500\ m^2$. The building's minimum, maximum, and average load are 2.7 kW, 25 kW, and 10.8 kW, respectively. 30 vehicles were chosen as fleet size in accordance with the example in [86]. To charge the vehicles, 30 charging points of 4.6 kW were selected. PV was assumed to cover 10% of the rooftop area with a power density of $175\ \frac{W}{m^2}$, equalling an install capacity of $8.75\ kW$. The maximum and average power output are 7.3 kW and 1.1 kW, respectively. Since a load of around 80 kW would occur in the case of charging half of the fleet at max capacity during peak building load,

80 kW was chosen for the grid connection capacity. Figure 5.1.3 shows the building's load profile and PV output.



(a) Building load

(b) PV power output

Figure 5.1.3: Building load and PV output for caretakers

Apart from their limited ability to invest, caretakers view range anxiety as one of their main concerns. This, however, can be mitigated by feasibility studies such as *SMobilityCOM*, which have yielded promising results for the sector [85]. Further constraints might be posed by limited grid connection capacities, as caretaker company sites were not designed for high electrical loads.

Utility companies

Utility companies own fleets of service vehicles to conduct installations or maintenance, either at the customer or in the field to service equipment [88]. Looking at medium-sized utility companies, a geographical focus exists, and it is common that the customer base of a city is distributed among several utilities. These companies therefore cover short to medium distances on a regular basis and are well-suited for fleet electrification, according to the Head of Real Estate and Facility Management at Vattenfall Berlin [89]. Weekend operation was assumed possible and the standard deviation of distance was assumed relatively high due to unplanned maintenance works. Table 5.1.6 shows the parameters for utility vehicle schedules.

Parameter	Mean	Standard deviation
Distance Weekday	120 km	30 km
Distance Weekend	80 km	25 km
Departure Weekday	07:00	1.0 h
Arrival Weekday	19:00	1.0 h
Departure Weekend	09:00	2.0 h
Arrival Weekend	16:00	2.0 h

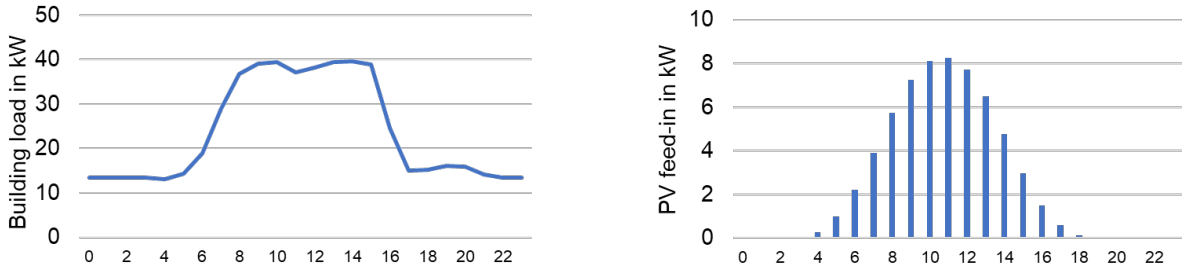
Table 5.1.6: Utility schedule parameters

The vehicles typically contain tools and equipment to perform the day-to-day activities. They require some cargo space, although not as extensive as was the case for last-mile delivery. Therefore, the Citroen e-Berlingo, a small- to medium-sized light duty vehicle was chosen [90]. Its parameters can be found in Table 5.1.7.

Citroen e-Berlingo	**Value**	**Unit**
Range	250	km
Battery capacity	50	kWh
Engine power	100	kW
Charging power	7.4/43	kW AC
Consumption mean	0.23	kWh/km
Consumption std. deviation	0.155	kWh/km
Consumption min	0.193	kWh/km
Consumption max	0.453	kWh/km

Table 5.1.7: Utility vehicle specifications

The utility company both requires storage space to store meters and other spare parts, as well as office space to house employees working in customer service and management, which is why two reference buildings were combined: the warehouse and the small office. The warehouse was scaled to $2500\ m^2$ and the office to $500\ m^2$. The minimum, maximum, and average load are 5.4 kW, 65.5 kW, and 23.6 kW, respectively. 25 vehicles were chosen, making up a medium-sized fleet. The utility was assumed to have access to charging at 22 kW AC. PV was assumed on 20% of rooftop area, making up 100 m2 and 17.5kW. The maximum and average power output are 12 kW and 2.2 kW, respectively. A grid connection of 1000 kW was assumed for the utility, because of its likely proximity to high-capacity grid connections. Figure 5.1.4 shows the annual average for building load and PV.



(a) Building load

(b) PV power output

Figure 5.1.4: Building load and PV output for the utility use-case

With their relatively short trip distances, high grid connection capacities and sufficient

funding capabilities, utilities are well-suited for EV fleet electrification [89]. This also reflects in the sector's electrification rate, as shown in Figure 5.1.1.

Comparative analysis

Figure 5.1.5 compares the three use-cases by their SOC upon arrival after a trip. All of them show skewed distributions with means between 20% and 40% SOC. Given a target SOC of 85%, this corresponds to a charging demand of ca. 45% to 65% per trip.

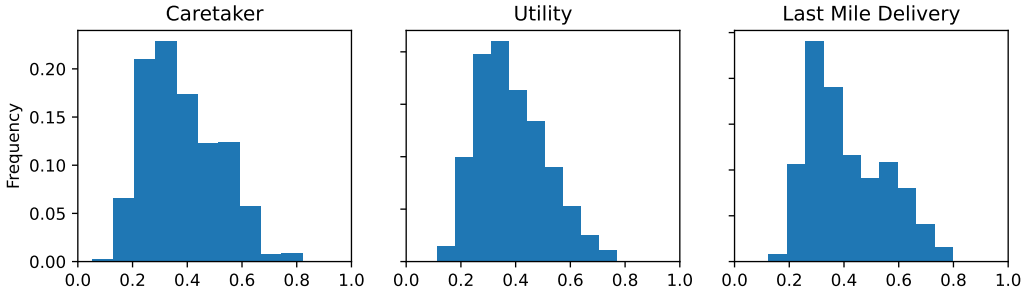


Figure 5.1.5: SOC upon arrival per use-case

Figure 5.1.6 compares the use-cases and scenarios by charging energy demand and annual charging cost. Due to the different number of cars, battery sizes and schedules, the charging energy demand and annual charging cost vary significantly across the use-cases. Charging cost was computed by taking the average of 2021 prices - either without or with markups.

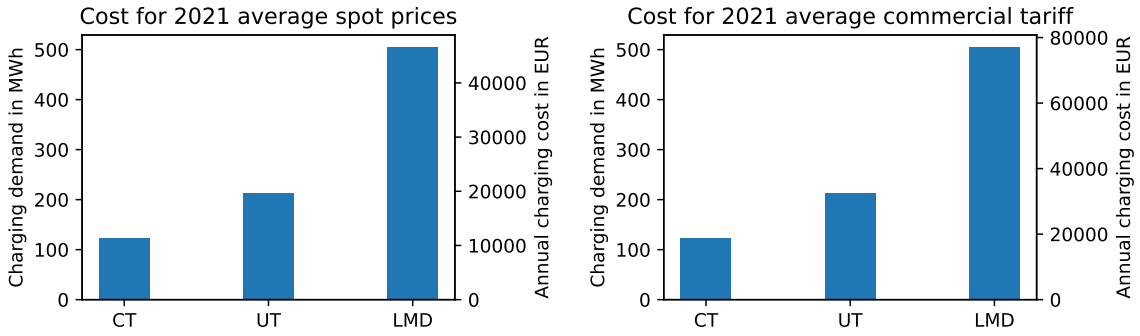


Figure 5.1.6: Comparison of charging energy and cost across use-cases and scenarios

Figure 5.1.7 visualizes the arrival and departure distributions per use-case. While utility and last mile delivery have similar shapes, the caretaker distinguishes itself with the arrivals during the day due to the lunch break.

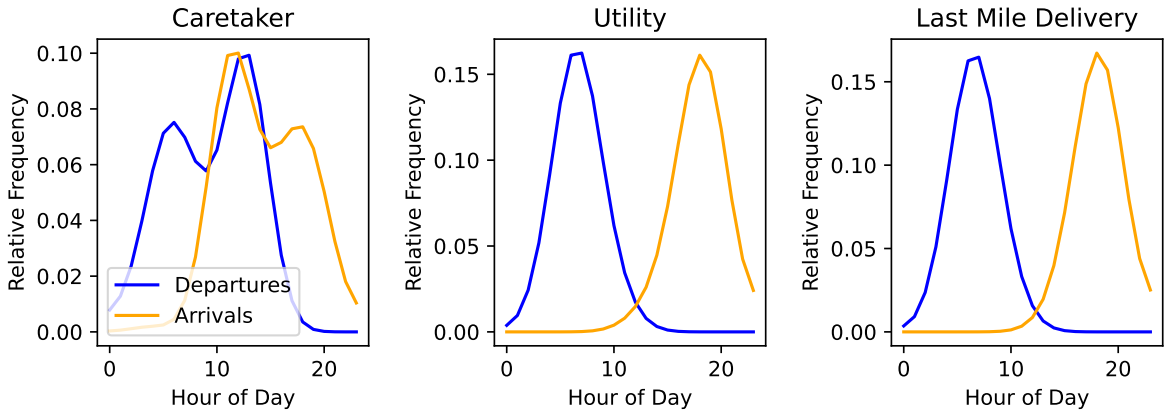


Figure 5.1.7: Departure and arrival distribution per use-case

5.2 RL-based optimization

In order to optimize EV charging schedules with RL, the problem must be implemented as an MDP. First, available implementations are reviewed, alongside with presenting an own implementation to tackle the charging problem. Second, the physical system is explained: the charging model, electricity prices, and battery degradation. Third, the MDP model is presented with its action and observation space, as well as reward function. The agent training process is explained in the fourth subsection. Finally, the method for evaluating the agents is explained.

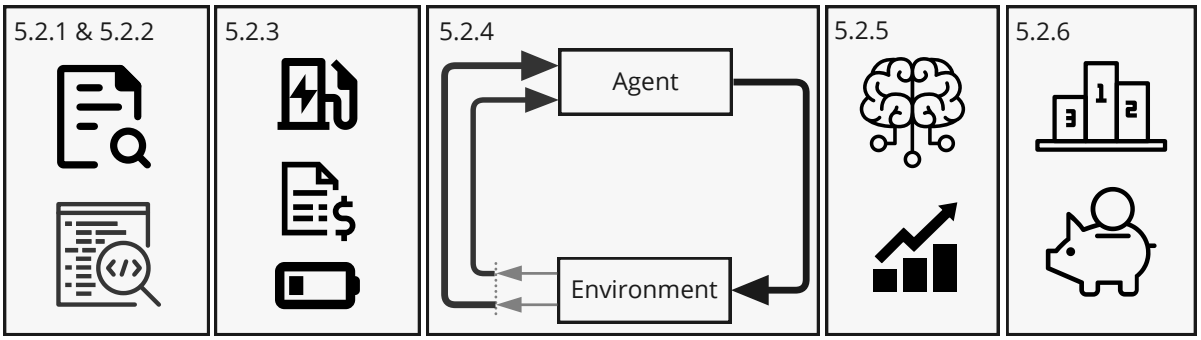


Figure 5.2.1: Structure of the optimization approach

5.2.1 Existing RL-based optimization frameworks

In a first step, a review was conducted of already existing implementations of the EV charging problem. Although the application of RL in EV charging optimization has gained significant scientific attention in recent years, there is a lack of publicly available

implementations. Nevertheless, three published frameworks have been found: *ACN-Sim*, *Chargym*, and *MAS* [91–93]. A brief review of the three frameworks is given below.

ACN-Sim stands for adaptive charging network and models the electrical network of real-world sites in the US, enabling simulations regarding the impact of EV charging on the grid. The implementation of RL-based optimization was thereby only implemented as an add-on to the framework, and the main focus was set on grid simulations. While the framework provides an impressive scope of features, it heavily revolves around the geographical focus of the US and the scope of the use-case it was implemented in. Vehicle schedule datasets, as well as the layout of the charging sites, were designed for a specific use-case, thus making it difficult to apply the framework to other use-cases and data sources. Further, the degree of documentation and maintenance since the release in 2021 was rather limited and the framework now relies on outdated implementations of TensorFlow [94]. The framework was therefore not implemented in this study.

Chargym is one of the first publicly-available frameworks that focuses on the MDP-formulation of the EV charging problem, as well as the evaluation of different RL algorithms. Its objective is to minimize charging expenses. Featuring the control of multiple vehicles, PV, time-of-use-tariffs, and bidirectional charging, the framework already constitutes a strong foundation for development efforts in RL-based EV charging. However, it also comes with limiting assumptions that make it difficult to design realistic commercial use-cases. First, the framework's episodes are fixed in length, from 00:00 to 23:59, with an hourly resolution. Because the environment is reset after each episode, overnight charging is rendered impossible. Second, *Chargym* implements price and PV data via hard-coded curves instead of integrating recent market price or weather data. Since the assumptions and simplifications of the framework were engrained in the code's logic and difficult to change, *Chargym* was also not used in this study. It was, however, an extremely helpful framework for this study, as it provided a well-developed example of how a RL-based implementation for EV charging could look like.

MAS stands for multi-agent system and features a DQN-based implementation of the EV charging problem with integrated load forecasting via Long Short-Term Memory (LSTM). The framework originates from the *E-Balance Plus* project - a Horizon 2020 funded research project [93, 95]. *MAS* was developed with a close link to real-world

applications, and bidirectional chargers are currently being installed at the research site to test the software in the field. The framework's objective is to steer the charging process in such a way, that it follows a pre-defined load curve set by the Distribution Service Operator (DSO). Because the framework is based on DQN, it features a discrete action space that is specific to the designed use-case. Because this thesis is considering continuous actions, *MAS* was not chosen in this study. Nevertheless, the framework and the correspondence with the authors provided insights that were invaluable to this study.

5.2.2 Own implementation

Due to limitations and a lack of applicability to commercial fleet use-cases in existing frameworks found in literature, a novel framework was developed within the scope of this study: FleetRL.

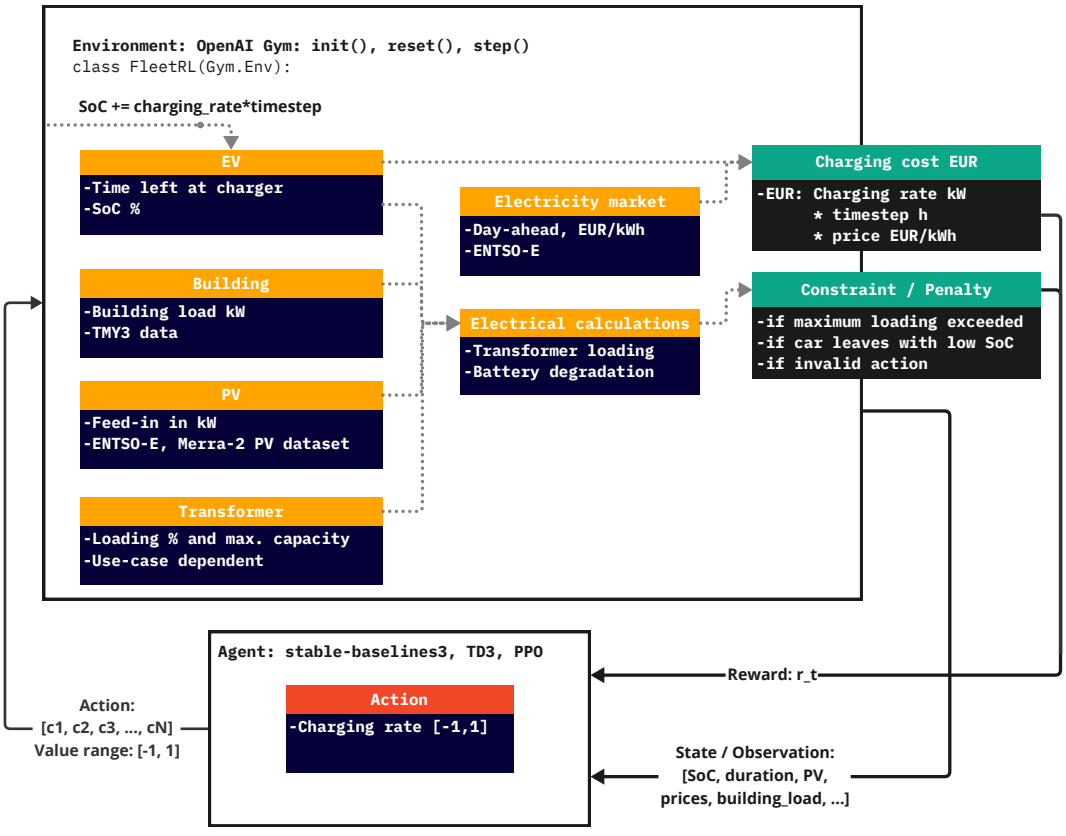


Figure 5.2.2: Basic conceptual overview of FleetRL

FleetRL was developed using a modular, object-orientated approach to allow for a high degree of customizability. This way, the same framework can easily be applied

to other use-cases, and integrate different methodologies and data sources. To model the EV charging problem within an RL environment framework, different classes were created that handle different aspects of the problem, e.g., communication with the RL agent, loading and processing of input data, calculating EV charging cost, or calculating battery degradation. Figure 5.2.2 provides an illustration of the framework's structure. Its flow of information is identical with the schematic of the MDP shown in Figure 4.2.1. The most important classes and aspects of the framework are described in the appendix, and the entire codebase is available under [9].

5.2.3 Modelling scope

The first step consists of defining the physical scope and boundaries of the real-world problem, so that it can be broken down into its sub-components and fundamental data structures. In this study, the goal is to optimize EV charging schedules by cost, while considering vehicle schedules, building load, PV generation, grid constraints, and electricity prices. This is illustrated in Figure 5.2.3 for the use-case of last-mile delivery.

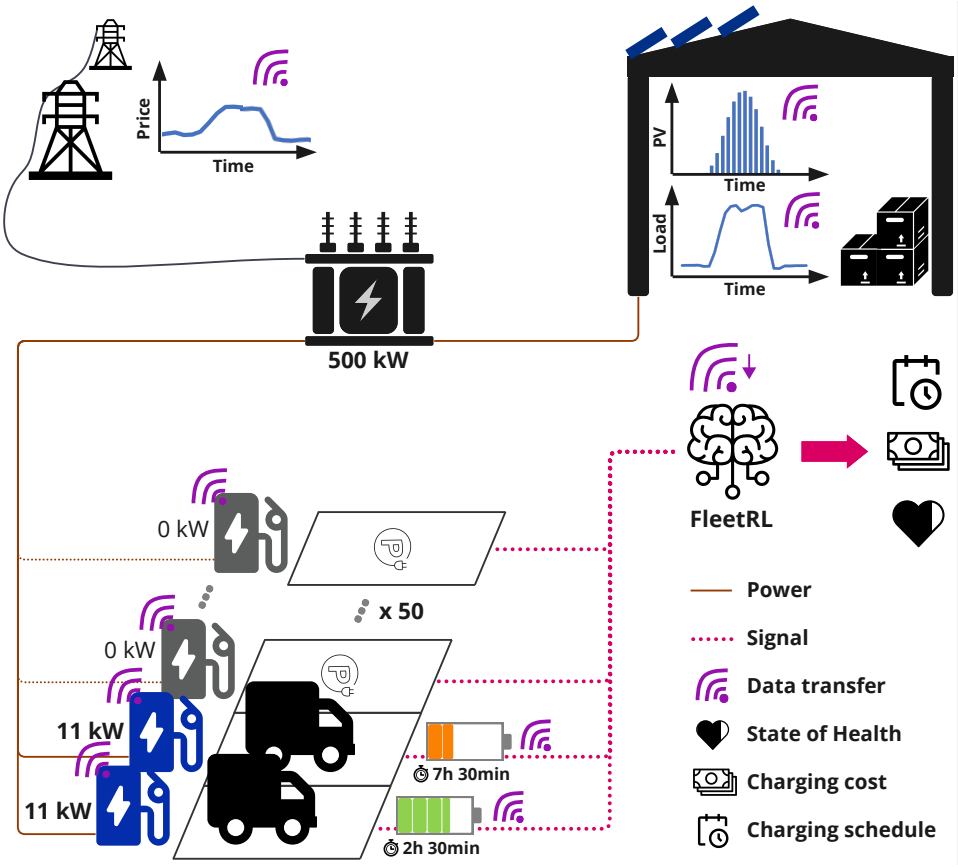


Figure 5.2.3: Scope of the EV charging problem

Some assumptions are taken to limit the degree of complexity of the optimization problem:

1. All connections are three-phase, and load imbalance is not considered.
2. Only active power is considered when calculating transformer loading. Capacitive and inductive properties of electrical equipment are not considered.
3. Although bidirectional charging is enabled, it only allows for energy arbitrage. Frequency regulation is not implemented in this study.
4. The companies are modelled as price takers and are assumed not to influence the electricity price with their demand.
5. Battery degradation is modelled non-linearly, taking into account rainflow cycle-counting and SEI-film formation according to [96].

Charging model

Bidirectional charging is enabled to explore potential effects on economics and battery degradation. The following function is implemented:

$$SOC_{t+1} = \begin{cases} SOC_t + \frac{E_{charging} \cdot \eta_{charging}}{C_{battery}} & , action \geq 0 \\ SOC_t - \frac{E_{discharging}}{C_{battery}} & , action < 0 \end{cases} \tag{5.1}$$

When discharging, the efficiency is not considered in the decrease of the SOC. Instead, it is considered when calculating the usable output of the battery that is sold on the spot market. Charging or discharging energy are calculated based on the agent's action, the available charging power and the amount of energy left in the battery. Actions that would yield an SOC > 1 or < 0 are clipped, and the agent is penalized by a small amount. When available, PV reduces the total energy drawn from the grid (self-consumption). Excess PV energy that is not used by the EVs contributes to covering the building load.

Electricity consumption tariff

Dynamic electricity tariffs were implemented to study the agent's ability to develop an optimal charging strategy that leverages intraday price differences. Price data from 2020 is not the most recent and was impacted by the pandemic. The year 2022 showed

unprecedented price spikes due to the European energy crisis. For this reason, spot prices for the market area of Germany / Luxemburg from 2021 were considered in this study, as shown in Fig. 5.2.4 [12]. That being said, this study focuses on the methodology of RL-based EV charging optimization and on investigating if an RL agent can learn an optimal policy within a realistic environment. If a specific use-case with different price assumptions is to be chosen, they can be incorporated by simply changing the input file to the model.

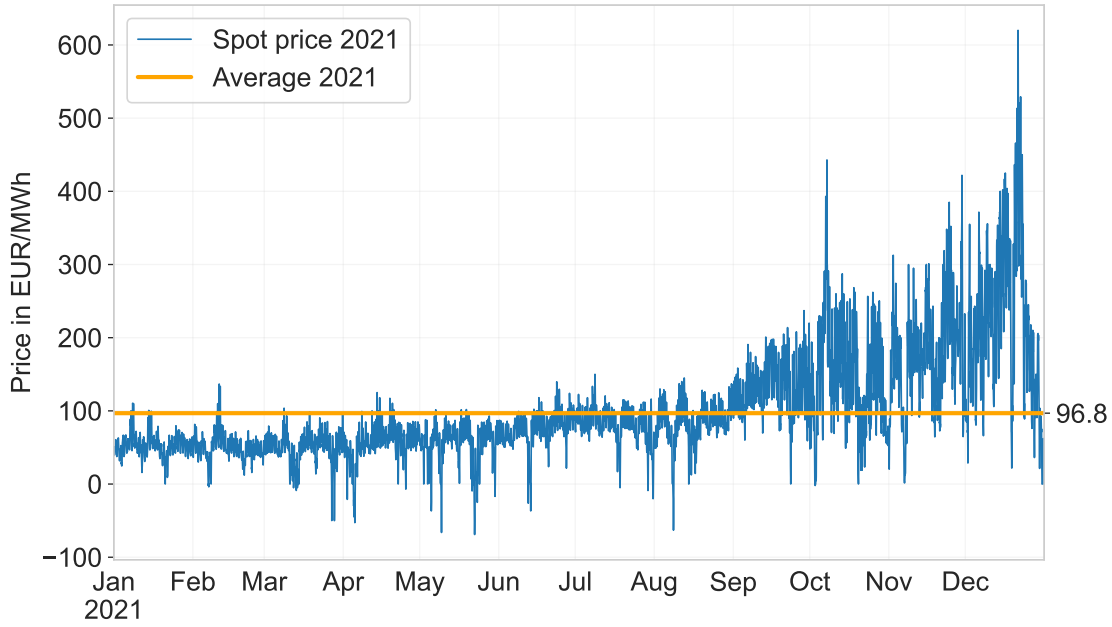


Figure 5.2.4: Day-ahead spot price DE-LU for 2021 [12]

According to Awattar and Tibber, two energy providers offering dynamic pricing in Germany, fees make up ca. 50% of the electricity price [22, 23]. This is in line with industrial electricity prices, where fees made up and 52% in 2021 [97]. Therefore, a constant fee (e.g. grid or concession fees) and a variable fee (e.g. electricity tax) were added to the spot price to simulate prices paid by commercial customers.

$$P_{total,avg,2021} = (96.8 + 10) \cdot 1.5 = 160.2€/MWh = 16.02ct/kWh$$

According to the German Association of Energy and Water Industries (BDEW), prices paid by industrial customers amounted to $21.38c/kWh$ for 2021 [97]. This is higher

than the modelled prices, because hourly time-of-use tariffs are not commonly found for industrial consumers, where cost stability often outweighs the need to leverage flexibility. Industrial customers are separately charged by their capacity and their energy consumption in Germany. The highest quarter-hourly energy demand in kW thereby determines the peak load. There is therefore a strong incentive to stay under a certain kW rating - which is why grid capacity limits were included in this model. Energy bills are issued regularly for the kWh consumed. Tariffs may vary per month, but usually have fixed rates. Values for grid and concession fees may vary significantly across different regions.

Electricity feed-in tariff

The agent can choose to discharge energy from the batteries to the grid. Since vehicle-to-grid is still in its early stages in Germany, it was not possible to find commercial tariff packages. In this study, two scenarios are assumed: energy arbitrage on the spot market and a feed-in-tariff; frequency reserve was not considered.

In the arbitrage scenario, the company is assumed to be an independent power producer that has access to the spot market. This scenario aims to maximize the potential of the vehicles' batteries within their mobility constraints. The company can trade electricity bidirectionally, and receive the spot price both for charging and discharging without markups or fees. Taxes are deducted in a post-processing stage.

In the second-scenario, a feed-in tariff similar to PV feed-in was chosen. As of 2023 and according to the German renewable energy law (EEG §48 Abs. 6), PV panels receive between $6.2ct/kWh$ and $8.6ct/kWh$, with decreasing tariffs for higher kWp ratings [98]. For all commercial use-cases, the cumulative peak power from the chargers surpasses $100kWp$, which is why $6.2ct/kWh$ were chosen as the feed-in tariff. From this, 25% were deducted to account for third party fees (e.g. metering and handling fees).

Computing hourly averages over the entire year yields valuable insights into the charging economics, as shown in Figure 5.2.5. Intraday fluctuations become evident, with the common troughs at night and noon, as well as the morning and evening peaks. Further, it can be seen that the PV-based feed-in tariff is insufficient to provide a feasible business case for selling electricity. While the tariff is beneficial for PV owners who generate electricity from solar energy, it will always yield losses when the energy

is first purchased from the spot market with fees and markups. Note that charging efficiencies further decrease the economics of arbitrage, since a part of the usable energy is lost in the charging and discharging process. For the feed-in scenario, it was therefore expected that the agent will choose not to discharge, but to postpone the charging to the early morning hours when electricity prices are lower.

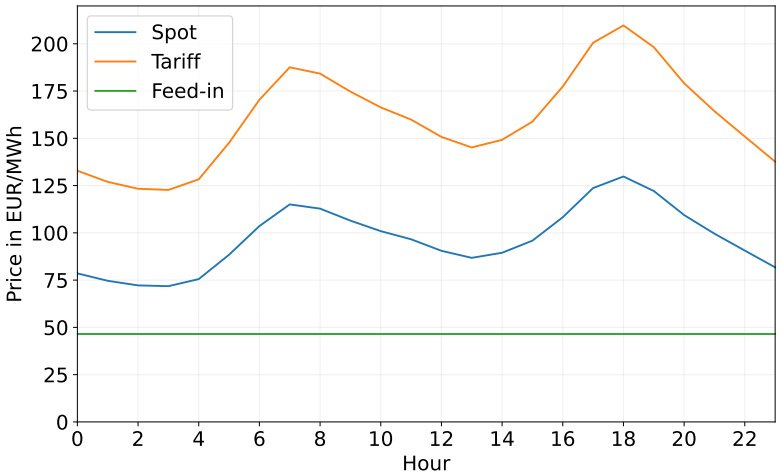


Figure 5.2.5: Annual average curve for 2021 electricity prices and feed-in tariff

Battery degradation

Two ways of calculating battery degradation were implemented: linear and non-linear degradation. Battery degradation was implemented such that the model updates the battery capacity on a daily basis while the model is running. The vehicles' batteries therefore degrade in real-time during the training process.

Linear degradation is implemented for the E63 Li-Ion battery cell manufactured by LG Chem [99]. Using the graphs provided in the data sheet under section A1.2 and A2, it was possible to determine the capacity fade in the battery per equivalent full cycle and per unit time, as shown in Table 5.2.1.

Parameter	Value
Cycle loss at 11 kW	0.0125% per EFC
Cycle loss at 22 kW	0.0125% per EFC
Cycle loss at 43 kW	0.0167% per EFC
Calendar ageing at 0% SoC	0.65% per a
Calendar ageing at 40% SoC	2.93% per a
Calendar ageing at 90% SoC	6.5% per a

Table 5.2.1: Linear degradation parameters

Non-linear degradation was modelled according to [96]. Apart from non-linear relationships, the model incorporates the formation of the solid electrolyte interphase: a thin layer that is formed during the first cycles of a battery [100–102]. Because Lithium Ions are consumed to form this film, the degradation of a battery progresses steeper during the period of SEI-film formation [96, 103, 104].

The governing equations of the model are shown below [96]. Stress factors determine the stress of each cycle in terms of temperature T, mean SoC σ, time t, and DoD δ. Via the stress factors, the degradation due to cycle and calendar ageing can be determined and aggregated to f_d^{cyc}. Summing over every cycle's aggregated degradation yields the total degradation f_d, which is used to calculate the battery life L. The battery life L thereby is the inverse of the state of health.

$$S_T(T) = e^{k_T(T-T_{ref})\cdot\frac{T_{ref}}{T}} \tag{5.2}$$

$$S_\sigma(\sigma) = e^{k_\sigma(\sigma-\sigma_{ref})} \tag{5.3}$$

$$S_t(t) = k_t t \tag{5.4}$$

$$S_\delta(\delta) = (k_{\delta 1}\delta^{k_{\delta 2}} + k_{\delta 3})^{-1} \tag{5.5}$$

$$f_c(\delta, \sigma, T_c) = S_\delta(\delta)S_\sigma(\sigma)S_T(T_c) \tag{5.6}$$

$$f_t(t, \sigma, T_c) = S_t(t)S_\sigma(\sigma)S_T(T_c) \tag{5.7}$$

$$f_d^{cyc}(\delta, t, \sigma, T_c) = f_c(\delta, \sigma, T_c) + f_t(t, \sigma, T_c) \tag{5.8}$$

$$f_d = \sum_{cyc} f_d^{cyc}(\delta, t, \sigma, T_c) \tag{5.9}$$

$$L = 1 - \alpha_{sei}e^{-\beta_{sei}f_d} - (1 - \alpha_{sei})e^{-f_d} \tag{5.10}$$

$$SoH = 1 - L \tag{5.11}$$

In the study, the parameters were fitted to empirical tests of a Li-Ion battery cell. The parameters were adopted and are listed below [96].

Parameter	Value	Explanation
α_{sei}	5.75%	Charge consumed during SEI film formation
β_{sei}	121	SEI model coefficient
$k_{\delta 1}$	140000	DoD stress factor coefficient
$k_{\delta 2}$	-0.501	DoD stress factor coefficient
$k_{\delta 3}$	-123000	DoD stress factor coefficient
k_{σ}	1.04	SoC strecc factor coefficient
σ_{ref}	0.5	SoC reference point
k_T	0.0693	Temperature stress factor coefficient
T_{ref}	298.15 K	Reference temperature
T	298.15 K	Ambient temperature assumed in FleetRL
k_t	4.14e-10 s^{-1}	Calendar ageing stress factor coefficient

Table 5.2.2: Non-linear degradation model parameters

As for the inputs of time, DoD, and SoC per cycle, the rainflow cycle-counting algorithm according to ASTM E1049-85 was used [105, 106]. The algorithm receives a signal as input - in this case the historical values of SoC - and outputs a range, mean, count, start, and end index. The range is equal to the DoD δ and the mean is equal to the average SoC σ. The count is either 0.5 or 1 and acts as a measure of severity for the respective cycle. The rainflow counting algorithm therefore takes into account that cycles differ in severity depending on their profile. Via the start and end index, the duration of the cycle can be calculated. These values are then fed into the non-linear degradation model.

Figure 5.2.6 compares the two methods of degradation calculation. Over a period of 10 years, the two methods reach approximately the same level of degradation, reaching the end of their life at a remaining SoH of 80%. However, the linear model progresses more steeply than the non-linear model after year 3 and surpasses it in year 8. If the time horizon were to be increased further, the linear model would cross the x-axis and reach negative SoH values. The assumption of linear degradation is therefore a simplifying one, which is only valid in a certain range of data points. To model battery degradation more accurately, the non-linear model is used in the rest of this study. Note that both models do not model the steep decrease of State of Health (SOH) which occurs towards the end of life of a battery [96]. An implicit assumption is therefore taken that the battery is operated within ranges of stable degradation, usually defined at SOH values between 0.8 and 1.

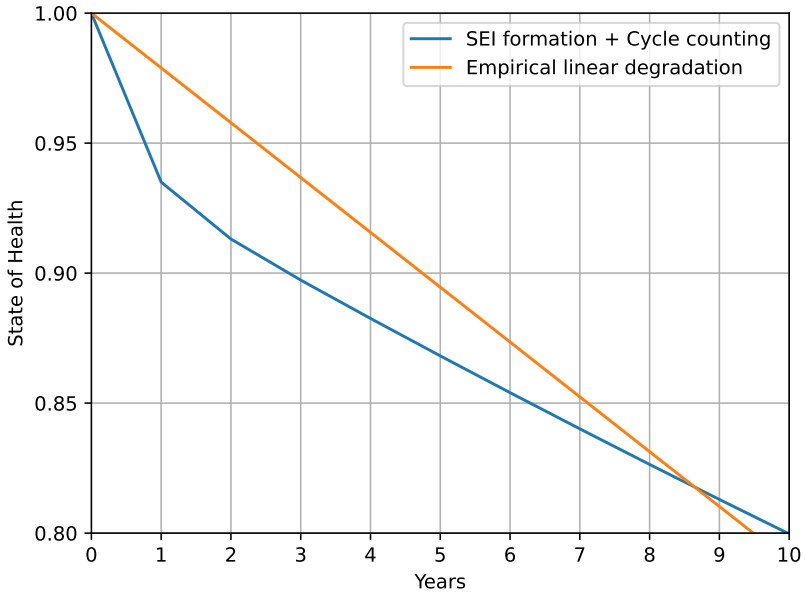


Figure 5.2.6: Linear vs. non-linear degradation over 10 years

5.2.4 Markov decision process

In order to be able to optimize EV charging with RL, the problem needs to be formulated as an MDP. As shown in Figure 4.2.1, the MDP describes the flow of information between the environment and the agent. The environment encodes the entire information of the EV charging problem, processes the RL agent's actions, and returns the next observation and reward. The agent then takes the observation as input and returns corresponding actions, which aim to maximize the cumulative reward.

Actions, Observations and Rewards

With the physical scope of the problem defined, the next step is the design of the observation and action space, as well as the reward function. The observation space defines the set of parameters that are visible to the agent. The action space defines the set of possible actions that are available to the agent. The reward function provides feedback to the agent after each action taken. It is important to define these metrics such that the Markov-assumption is met: the most recent observation must be a complete representation of past states and actions. The following setup is chosen for this study:

Action space:

- Charging power: $[C_{EV_1}, C_{EV_2}, ..., C_{EV_N}] \in [-1, 1]$

Observation space:

- State of Charge: $[SOC_{EV_1}, SOC_{EV_2}, ..., SOC_{EV_N}] \in [0, 1]$
- Time left at the charger: $[t_{EV_1}, t_{EV_2}, ..., t_{EV_N}] \in [0, t_{max}]$
- Spot price and tariff with 8 hours look-ahead: $[P_t, P_{t+1}, ..., P_{t+8}] \in [P_{min}, P_{max}]$
- Building load with 4 hours look-ahead: $[B_t, B_{t+1}, ..., B_{t+4}] \in [B_{min}, B_{max}]$
- PV with 4 hours look-ahead: $[PV_t, PV_{t+1}, ..., PV_{t+4}] \in [PV_{min}, PV_{max}]$

Both observation and action space are continuous, making the problem complex due to the necessity to perform function approximation, as discussed in Chapter 2. The look-ahead of PV and building load is assumed to be known. In future work, the use of future values could be replaced with deep-learning-based forecasting. As for the day-ahead price, the next 8 hours are always known since the price of the next 24 hours is published at noon [107].

Optional additions to the observation space

- Target SOC for each vehicle
- Remaining charging demand in %
- Time needed to fulfil charging demand
- Laxity factor: $\frac{t_{left}}{t_{needed}} - 1$
- Charger power in kW
- Grid connection in kW
- Available grid connection capacity
- Maximum recommended action per car: $\frac{P_{grid,avail}}{N_{EV,connected} \cdot P_{charger}} \in [0, 1]$
- Month, weekday and hour

The option to include additional information into the observation space was added to reduce the complexity of logical deduction needed by the agent. While current SOC and time left are sufficient to physically describe the basic problem, it might be difficult for the agent to infer the right charging strategy from these parameters. This is especially

true for model-free agents that are used in this study, because no prior information is given about the real-world problem. Therefore, seemingly redundant information can increase the training performance in a neural network. To highlight this, the training process for two last-mile delivery vehicles on the TD3 algorithm is shown in Figure 5.2.7; with and without auxiliary information.

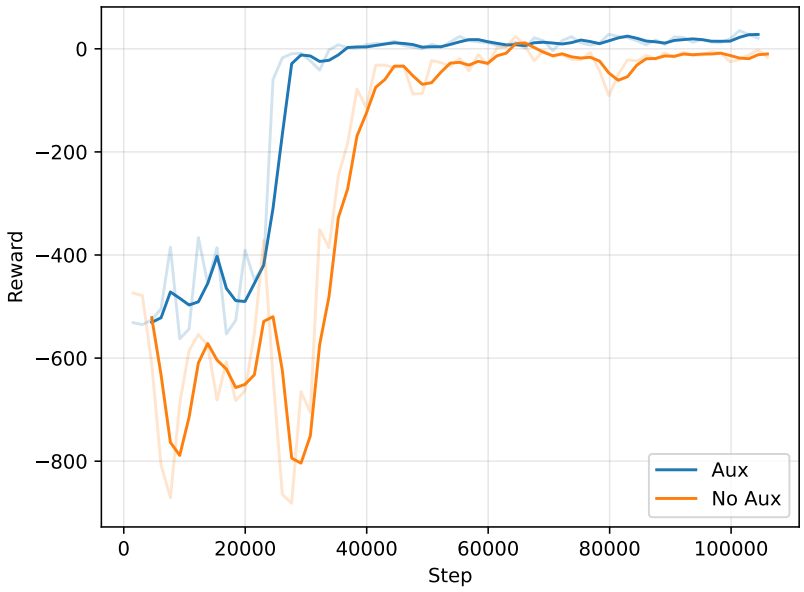


Figure 5.2.7: Impact of additional information on training

Reward function: The reward function is both made up of the charging cost and penalties when constraints are violated, or when invalid actions are taken. C_i denote scaling factors for further adjustment of the reward function.

- Charging cost: $R_1 = E_{charging} \cdot P_t \cdot C_{R_1}$
- Penalty for not meeting target: $P_1 = C_{P_1} \cdot sigmoid(SOC_{target} - SOC_i)$
- Penalty if grid overloaded: $P_2 = C_{P_2} \cdot sigmoid(Load_t - Load_{max})$
- Penalty for taking an invalid action: $P_3 = C_{P_3} \cdot action_{EV_i}^2$
- Penalty for overcharging the battery: $P_4 = C_{P_4} \cdot E_{overcharged}^2$

For the SOC and the grid violation, a sigmoid function is proposed, as shown in Figure 5.2.8

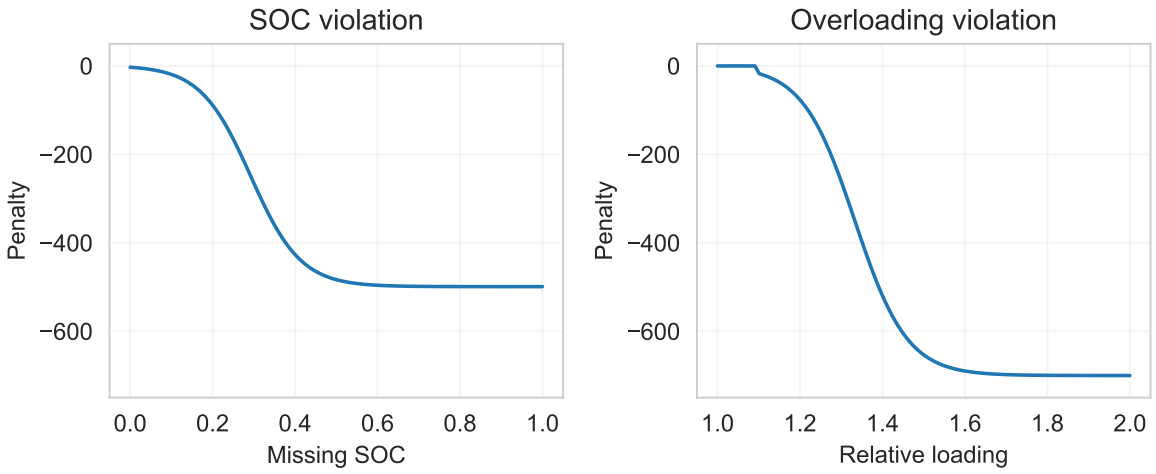


Figure 5.2.8: Sigmoid-based penalty curves

A sigmoid function was chosen because of its close similarity to the actual consequences in reality: Small SOC violations up to 5-10% do not significantly endanger the operation. Similarly, transformer loadings of 110% can be endured without risking damage to technical equipment [source]. With increasing violations, the risk of endangering operations increases over-proportionally. At some point, a level is reached beyond which the damage does not further increase, e.g. a blown fuse, damaged equipment, or a failed trip due to lack of energy in the battery. Note that the dip and its magnitude in the overloading curve can be set to custom values, e.g. a kW rating that must not be exceeded at all costs. This could avoid blowing a fuse, or exceeding a kW-peak rating for economic reasons.

Markov assumption

The representation meets the Markov criterion, because the agent's past actions of charging the vehicles are represented in the state of charge. Information regarding price, PV and building load is incorporated in the reward function. Further, the state representation and reward function are designed in coherence with the current state-of-the-art which is presented in Chapter 2.

Normalization and reward shaping

All observations and rewards are normalized between 0 and 1 via a rolling average according to Equation 5.12 to facilitate the training process.

$$obs_{normalized,i} = \frac{obs_i - obs_{min}}{obs_{max} - obs_{min}} \tag{5.12}$$

The reward function can be fine-tuned during training by giving certain components more weight than others via the scaling factors C_i. Apart from the economics, the reward relationships are quadratic to penalize higher deviations more severely.

The scaling factors are chosen such that the reward function distinguishes primary, secondary and tertiary objectives. The primary goal is to charge the vehicles in such a way that the SOC and grid constraints are met. Once this is achieved, the focus should be set on optimizing for cost. Finally, the tertiary objective of the agent is to take actions that make sense. Therefore, a penalty is given when a charging command is given for an empty charging spot. The three hierarchic levels are separated by an order of magnitude: violating the SOC or grid constraints therefore yields a penalty 10 times higher than the cost of electricity for fully charging a vehicle. The economic expense, on the other hand, is 10 times higher than the penalty for invalid actions or giving commands that would overcharge the battery. Due to this hierarchy, a rolling normalization is useful, because the numeric scale of the rewards does not change for the agent. Without normalization, the second and third tier penalties would appear insignificant, and would potentially be ignored.

Another important operation in the area of reward shaping was detrending the price signal before using it as a reward. As seen in Figure 5.2.4, it is possible that the general level of prices can change throughout the year. If rewards were to be directly computed from such a signal, higher or lower average prices would send a penalizing or rewarding signal to the agent. This can be counterproductive when the real aim is to learn how to exploit price *differences*. Therefore, the price signal was detrended by splitting the data into monthly chunks and offsetting each chunk's data, such that the chunk's average was equal to the annual average. This is illustrated in Fig 5.2.9 for the year 2021. Note that while the approach was sufficient for this specific set of data, it could be worth looking into time series decomposition or linear regression to remove trends and periodic signals more robustly.

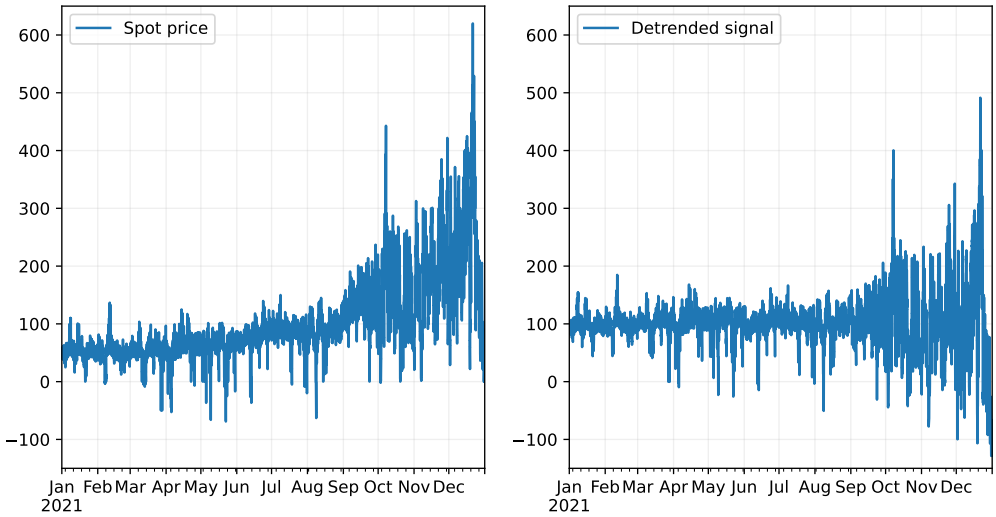


Figure 5.2.9: Detrended spot price for 2021

Additional model parameters

Apart from observations, actions, and rewards, a number of parameters fundamentally shape the model which are listed in Table 5.2.3. The episode length for training was chosen to be 48 hours to guarantee the possibility of overnight charging, regardless of the episode's starting time. The initial state of health was assumed to be 100%, therefore assuming a new fleet.

Parameter	Value
Temporal resolution	15 min
Episode length	48 hours
Charging efficiency	91 %
Discharging efficiency	91 %
Target SoC	85%
Initial SoH	100%

Table 5.2.3: FleetRL environment parameters

5.2.5 RL agent training

This section walks through the training process, from pre-processing to deploying agents in remote computing environments.

Data preprocessing

The input data was split prior to training to avoid overfitting the models. The data was split into training, validation, and test sets. During training, the agent was fed with the training set. At regular intervals during training, a run was performed on the validation set to assess the performance of the agent on unseen data of the same year. After concluding the training process, the agents were tested on an entirely new dataset to validate the results. Training and validation data were split into January - October (83%) and November - December (17%), respectively. For testing, a new set of schedules was generated.

Troubleshooting

At first, a large number of agents were deployed with different configurations to find out what would break the learning process or introduce instabilities. This was an iterative process and involved tuning the reward function and correcting logical flaws in the code that were revealed during the testing stage. It was found that the best configuration for the reward function was to split the objectives numerically into the following dimensions: $10^2 - 10^3$ (SOC and overloading the grid), 10^1 (fully charging the battery at non-negative prices), and 10^{-1} (invalid action and overcharging penalties).

Once a stable setup was reached, different agents were tested on the environment. The stable-baselines3 library was used for deploying RL agents on the environment [56]. Initially, TD3, PPO, DDPG, and SAC were tested. PPO and TD3 performed particularly well. While PPO was faster, it was less sample efficient and required a larger number of steps to converge, as shown in Figure 5.2.10.

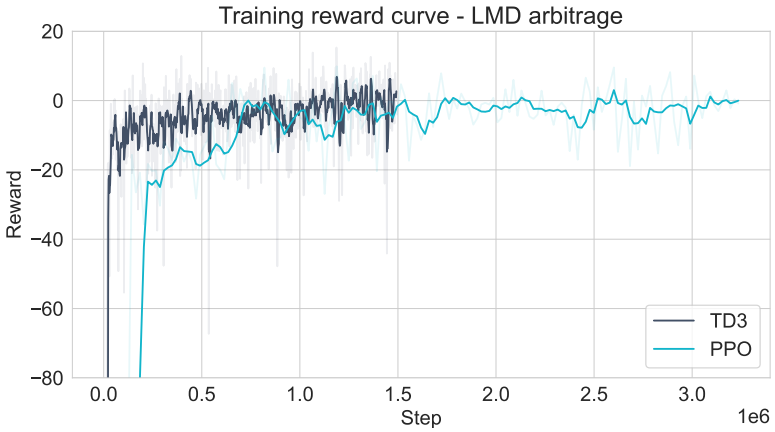


Figure 5.2.10: Performance comparison of TD3 and PPO

Although TD3 only ran at half the speed, it reached similar rewards as PPO - even with 1.5M steps less of training. Since TD3 and PPO are state of the art model-free DRL algorithms, and because training resources were limited, most runs were performed on TD3 and PPO only.

Hyperparameter tuning

An extensive hyperparameter search was performed on TD3 and PPO via the open-source framework Optuna. The search spaces are shown in Tables 5.2.4 and 5.2.5.

Parameter	Possible values	Best value
Gamma	0.9 - 0.99	0.99
Learning rate	1e-5 - 1e-1	0.0005
Batch size	32 - 512	128
# Epochs	1e4 - 1e6	8
GAE lambda	1e-4 - 1e-2	0.9
Clip range	5e3 - 5e4	0.2
Clip range VF	None - 0.5	None
Normalize advantage	Yes / No	Yes
Entropy coefficient	0 - 1e-3	0.0008
VF coefficient	0.1 - 0.7	0.5
Max gradient norm	0.3 - 0.7	0.5
# Steps	128 - 8192	2048

Table 5.2.4: Hyperparameters PPO

Parameter	Possible values	Best value
Gamma	0.9 - 0.99	0.99
Learning rate	1e-5 - 1e-1	0.001
Batch size	32 - 512	100
Buffer size	1e4 - 1e6	1e6
Tau	1e-4 - 1e-2	0.01
Learning starts after	5e3 - 5e4	2e4
Training frequency	2 - 8 steps / episodic	4 steps / episodic

Table 5.2.5: Hyperparameters TD3

For further reference on the hyperparameters, the original publications should be considered, as well as the Stable-Baselines3 (SB3) documentation [56, 61, 68]. For the hyperparameter study, 100 runs were conducted on each algorithm, where each run evaluated the agent's performance after 50000 time steps. The results were critically analysed to ensure that the values are still within range of the default values suggested

by the authors. Note that for TD3, both a train frequency of 4 steps and 1 episode were found to be suitable configurations.

Curriculum learning

Curriculum Learning, as proposed by [108], is a training strategy that structures the learning process by progressively introducing more difficult tasks. In the context of RL, the tasks usually refer to increasingly complex versions of the same environment or problem [109]. Curriculum learning has been found to enhance learning efficiency and robustness by allowing the agent to develop necessary skills on simpler tasks before transitioning to more complex ones, ultimately aiding in the discovery of efficient learning policies [110]. It was tested to conduct curriculum learning on the environment, but during the brief trial period no significant performance increases were observed. It seemed particularly challenging to make models of differently-sized observation spaces (e.g., a different number of cars) compatible with each other. Curriculum learning was therefore not used in the training process, but identified as a future improvement step.

Training environment

The agents were trained on remote computing environments with graphical processing units. The training process was monitored with TensorBoard [111]. Up to 32 cores, 1 GPU, and 32 GBs of RAM were used per machine, with multiple machines running in parallel. Depending on the number of cars in the environment, training 1 million steps took between 24 and 96 hours. The process was time- and resource intensive and thus poses a potential bottleneck.

5.2.6 RL agent evaluation

With training completed, the agent performance was tested on the test set. Key performance indicators were the reward, the money spent on charging, the number of penalties triggered, and the level of battery degradation. Further, the actions were closely monitored time step by time step to ensure that the agent was actually solving the problem of EV charging without finding an unexpected loophole. The monitoring was implemented through print functions that output important metrics at each time step of an episode.

Benchmarking

The trained RL agents were compared to deterministic charging strategies to validate the results. Validation is particularly important for black-box optimization methods such as DRL, where the decision-making process cannot easily be retraced, if at all. Three static benchmarks are shown in Figure 5.2.11.

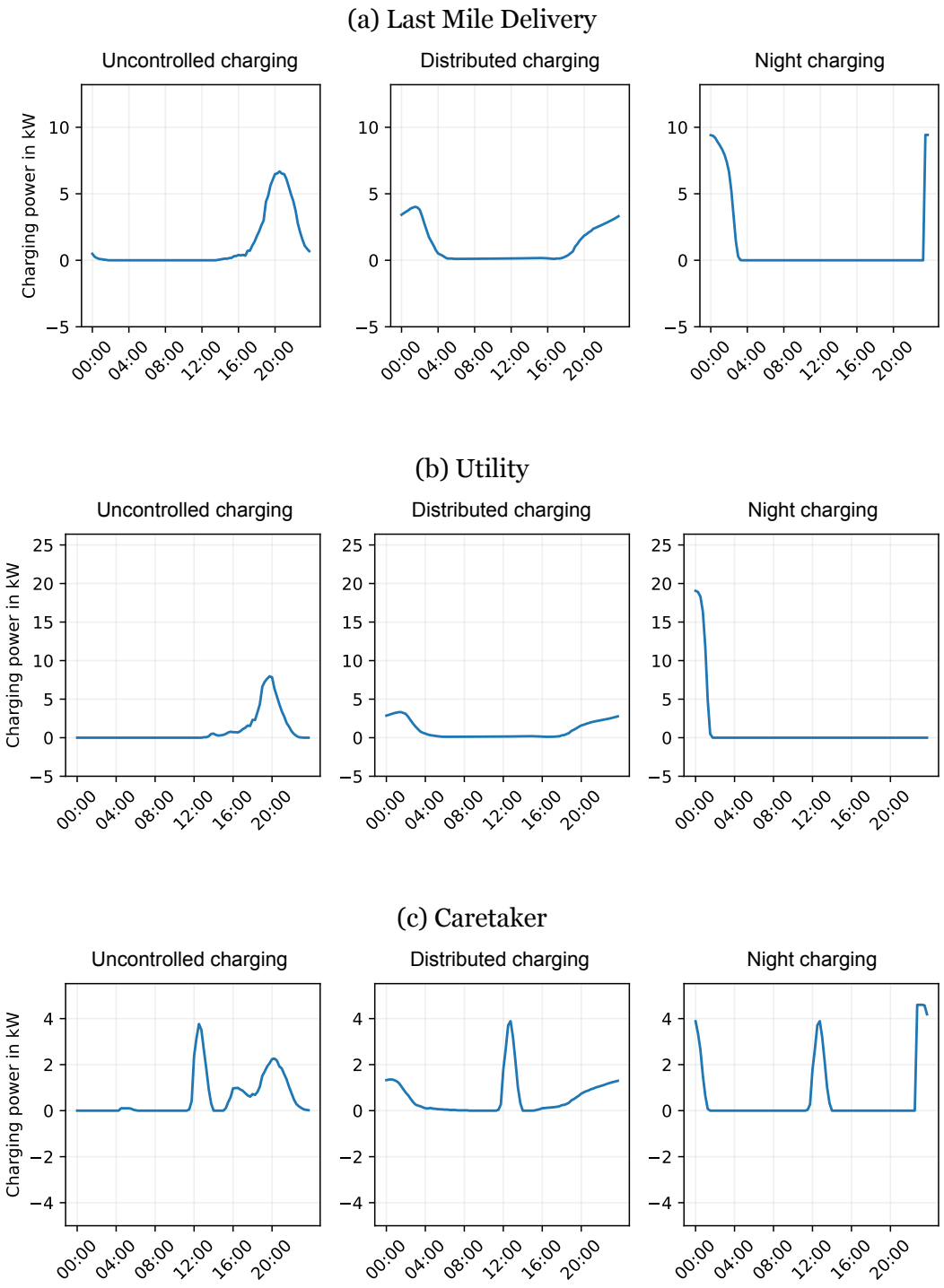


Figure 5.2.11: Static comparison benchmarks

The three rule-based strategies represent cases, where the user programs a fixed charging schedule into the EV. The methods were compared by key performance indicators such as monetary expenses, battery degradation, reward, and number of penalties triggered. The strategies are further explained below:

Uncontrolled / Dumb charging: The first comparison was made with the so-called *dumb charging* strategy, where an EV was immediately plugged in upon arrival, drawing the maximum power available at the charger.

Distributed charging: The second comparison was made with *distributed charging*, where the charging energy was spread out evenly during the charging duration. The cars were therefore fully charged only just before departure.

Night charging: A third comparison was made with *night charging*, where the vehicles only start charging at night. While leveraging lower prices, charging at was still started early enough to ensure that the battery was fully charged upon departure.

On top of the static benchmarks, a linear optimization was implemented to compare RL to an optimal scenario with perfect knowledge.

Linear optimization: A final comparison was made with a linear optimization model using Pyomo, in which the EVs optimize their charging schedules according to price, building load and PV. Note that the linear optimization benchmark was given perfect knowledge and was therefore expected to find the global optimum in terms of cost savings.

The model was given the parameters of building load, PV power output, electricity price, feed-in tariff, and EV availability. Similar to the RL agent, its decision variables were the charging and discharging power $\in [-1, 1]$. This made it possible to run the optimization result on the RL environment as if it were an agent, allowing for a more detailed analysis of the specific actions, as well as battery degradation, economics, and penalties. Pyomo and the Gurobi solver were used to run the optimization.

Parameters:

- Building load in kW B_t for $t \in [0, 8760]$
- PV output in kW PV_t for $t \in [0, 8760]$
- EV availability $EV_t \in [0; 1]$ for $t \in [0, 8760]$

- Price and Tariff P_t, T_t for $t \in [0, 8760]$
- SOC on return $SOC_{ret,t}$ for $t \in [0, 8760]$
- EVSE power in kW P_{EVSE}
- Timesteps per hours dt
- Charging and discharging efficiency η_C, η_{DC}
- Transformer capacity in kW P_{trafo}
- Battery capacity in kWh C_{batt}
- Target SOC SOC_{target}
- Initial SOC SOC_{init}

Decision variables:

- SOC $\in [0, SOC_{target}]$
- Charging signal C $\in [0, 1]$
- Discharging signal DC $\in [-1, 0]$
- Used PV for EV charging $PV_{used} \geq 0$
- Binary Positive action $A^+ \in [0; 1]$

Objective function:

$$C = \sum_{t=1}^{T} \left[\left(C_t * P_{EVSE} - PV_{used,t} \cdot \frac{1}{dt} \cdot P_t \right) + \left(DC_t * P_{EVSE} * \eta_{DC} \cdot \frac{1}{dt} \cdot T_t \right) \right] \quad (5.13)$$

Subject to:

- Max transformer loading: $(C_t + DC_t) \cdot P_{EVSE} + B_t - PV_t \leq P_{trafo}$
- Max charging: $C_t, DC_t \leq P_{EVSE} * EV_t$
- Mutual exclusivity charging: $C_t \leq A_t^+$
- Mutual exclusivity discharging: $DC_t \geq A_t^+ - 1$
- PV use: $PV_{used,t} \leq C_t \cdot P_{EVSE}$
- PV availability: $PV_{used,t} \leq PV_t$

- No charge at empty spot: $C_t, DC_t = 0$ if $EV_t = 0$
- SOC last timestep: $SOC_{t+1} = SOC_t + (C_t\eta_C + DC_t) \cdot P_{EVSE} \cdot \frac{1}{dt \cdot C_{batt}}$ if $t = T$
- SOC new arrival: $SOC_{t+1} = SOC_{ret,t+1}$ if $EV_t = 0$ and $EV_{t+1} = 1$
- SOC on departure: $SOC_t = SOC_{target}$ if $EV_t = 1$ and $EV_{t+1} = 0$
- Charging: $SOC_{t+1} = SOC_t + (C_t\eta_C + DC_t) \cdot P_{EVSE} \cdot \frac{1}{dt \cdot C_{batt}}$ if $EV_t = 1$ and $EV_{t+1} = 1$
- No next SOC when no car: $SOC_{t+1} = 0$ if $EV_t = 1$ and $EV_{t+1} = 0$
- No SOC when no car: $SOC_t = 0$ if $EV_t = 0$
- First SOC: $SOC_0 = SOC_{init}$

5.3 Economic impact assessment

A business case for RL-based EV charging is made based on the cost *savings* compared to the benchmarks. Therefore, it is assumed that EV chargers and EVs have already been procured; the economics of switching from internal combustion to electric vehicles are not considered. The economic impact assessment therefore only includes additional expenditures that would be necessary to implement RL-based charging, such as meters, communication devices, and third party fees. The cost data for ancillary equipment and fees was taken from the German Automobile Club (ADAC), which conducted a study on commercially available dynamic load management solutions [112]. These solutions, therefore, provide an integration of the EV charger with PV or the building load, enabling dynamic adjustments of the charging power whenever there is a constraint or price signal. Apart from performance comparisons, the study includes data on the installation and operational costs of dynamic load management systems. First, the annual spending on electricity for charging is compared against the benchmarks. Then, the profitability is assessed over a time horizon of ten years by calculating the break-even point and net present value of the investment. The assumptions for the economic calculations are listed in Table 5.3.1.

Parameter	Value	Unit
Discount rate	0.075	-
Load controller 2	1266	€
EVSE upgrade 4.6kW	300	€
EVSE upgrade 11 or 22 kW	500	€
Battery replacement 16.7 kWh	7500	€
Battery replacement 50 kWh	12500	€
Battery replacement 60 kWh	15000	€
Integration fee per charger	99	€
Third party service fees	0.1	share of savings

Table 5.3.1: Assumptions for economic calculations

Note that investments for a smart meter were not included in the calculation because, according to the German smart meter law, an installation is compulsory for buildings with a load greater than 6000 kWh or solar of 7 kWp and therefore does not depend on whether RL is used in smart charging or not.

Chapter 6

Results and discussion

This chapter presents and discusses the results. The first section focuses on general findings on algorithm performance that were obtained during the training and evaluation process. Then, detailed results are presented for each use-case: the charging strategies and the corresponding outcomes are analysed, and an economic case is presented for RL-based charging and its benchmarks.

RL-based EV charging performed well across all use-cases. This study therefore supports literature findings which claim that RL agents are a viable option for EV charging - even when using a realistic EV charging environment such as FleetRL that avoids the common simplifications of other RL-based EV charging frameworks (cf. Section 2).

6.1 Algorithm performance

Results were obtained for agents that simultaneously steered 1 EV and 5 EVs, after 5 million timesteps of training. All results are based on test sets: entirely new schedules that were not shown to the agent prior to testing.

PPO performed considerably better on all use-cases, regardless of the number of vehicles. On the same hardware, it was approximately twice as fast and managed to reach an acceptable policy quicker than TD3. Upon further inspection, it was also found that TD3 produced more SOC violations and yielded less economic savings in most of the runs. It particularly stood out that TD3 failed to learn not to discharge vehicles in the tariff scenario, while PPO gradually adapted its charging strategy

accordingly. TD3 therefore incurred high losses because it discharged energy at the low PV tariff, only to buy energy back more expensive later. Different hyperparameter configurations were tested for TD3 to try to avoid the issue. The most important hyperparameters were found to be the learning rate, the buffer size, and the training frequency. Updating TD3 once per episode seemed to improve the behaviour at first, but upon closer inspection it was found that sparse and instantaneous events, such as SOC violations, were then ignored more often. Changing the learning rate and buffer size often resulted in diverging or unstable agents. An optimal configuration was not found and due to limited time and computational resources TD3 was not used in the final set of runs that were evaluated in the coming sections.

Agents were trained for both 1 and 5 EVs. As shown in Figure 6.1.1, both agents converged, suggesting that agents can learn an optimal policy for multiple EVs if enough time and computational resources are allocated. Note that the curves were smoothed for visibility. For 1 EV, the training process took approximately 10 hours, while training with 5 vehicles at the same time took approximately 48 hours. Both runs used remote machines with 28 CPUs and 1 GPU, corresponding to ca. 40 Teraflops. Due to the computational intensity, it was not deemed feasible to train agents with 25-50 cars for this study. This might be attempted in future work, along with trying different MDP representations that do not scale with the number of vehicles.

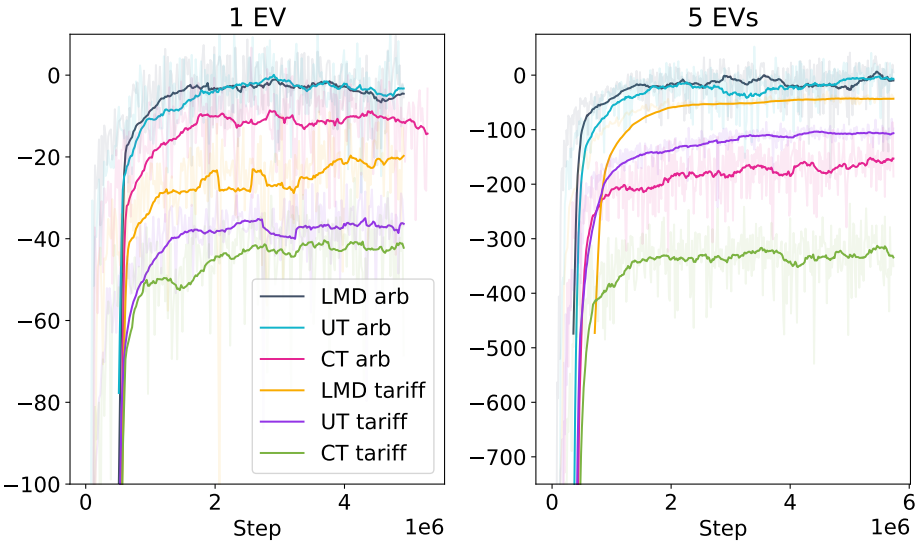


Figure 6.1.1: Smoothed reward function progression during training

Transitioning from 1 to 5 vehicles introduced challenges to the RL agent. First, the bigger observation and action space increased the complexity and made it more

difficult to infer an optimal charging strategy. This is particularly true for model-free agents such as TD3 and PPO: while adding more vehicles does not inherently change the dynamics of the problem, it poses a challenge to model-free agents that do not possess prior knowledge and thus do not know the dynamics of the problem. Having to learn an optimal policy from scratch is therefore likely to make the problem increasingly difficult with an increasing number of vehicles. Second, the problem dynamics varied slightly because 5 EVs can potentially overload a grid connection whereas 1 EV cannot. The agent therefore had to learn to stay within the limits of the grid connection in addition to charging 5 cars simultaneously.

However, evaluating the agent with 5 EVs showed that the training process was successful in that regard: overloadings either did not occur or remained at significantly lower levels compared to the static charging strategies. Further, SOC violations stayed within an acceptable range. This is illustrated in Figure 6.1.2.

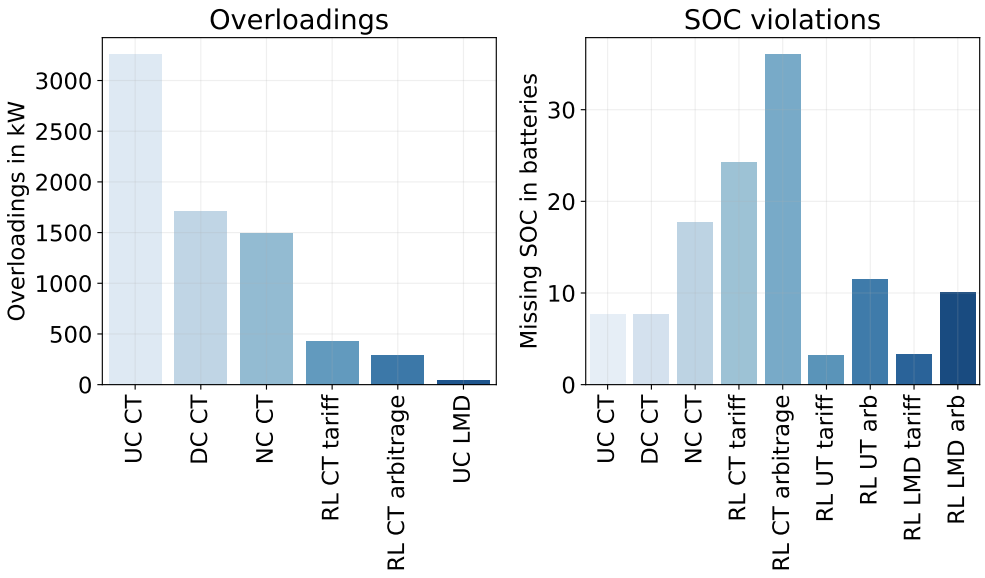


Figure 6.1.2: Overloading and SOC violations for 5 EVs over 1 year

A more detailed demonstration of the 5 EV agent is shown in Figure 6.1.3 for the caretaker use-case - the most tightly constrained in terms of grid connection. The RL agent is displayed for the arbitrage scenario, along with its static benchmarks: uncontrolled charging, distributed charging, and night charging. Differences in the charging strategies become apparent when inspecting the progression of charging power and SOC. Note that the SOC is an average of the 5 vehicles. SOC violations and grid overloadings are displayed. The caretaker's grid connection was 36.9 kW.

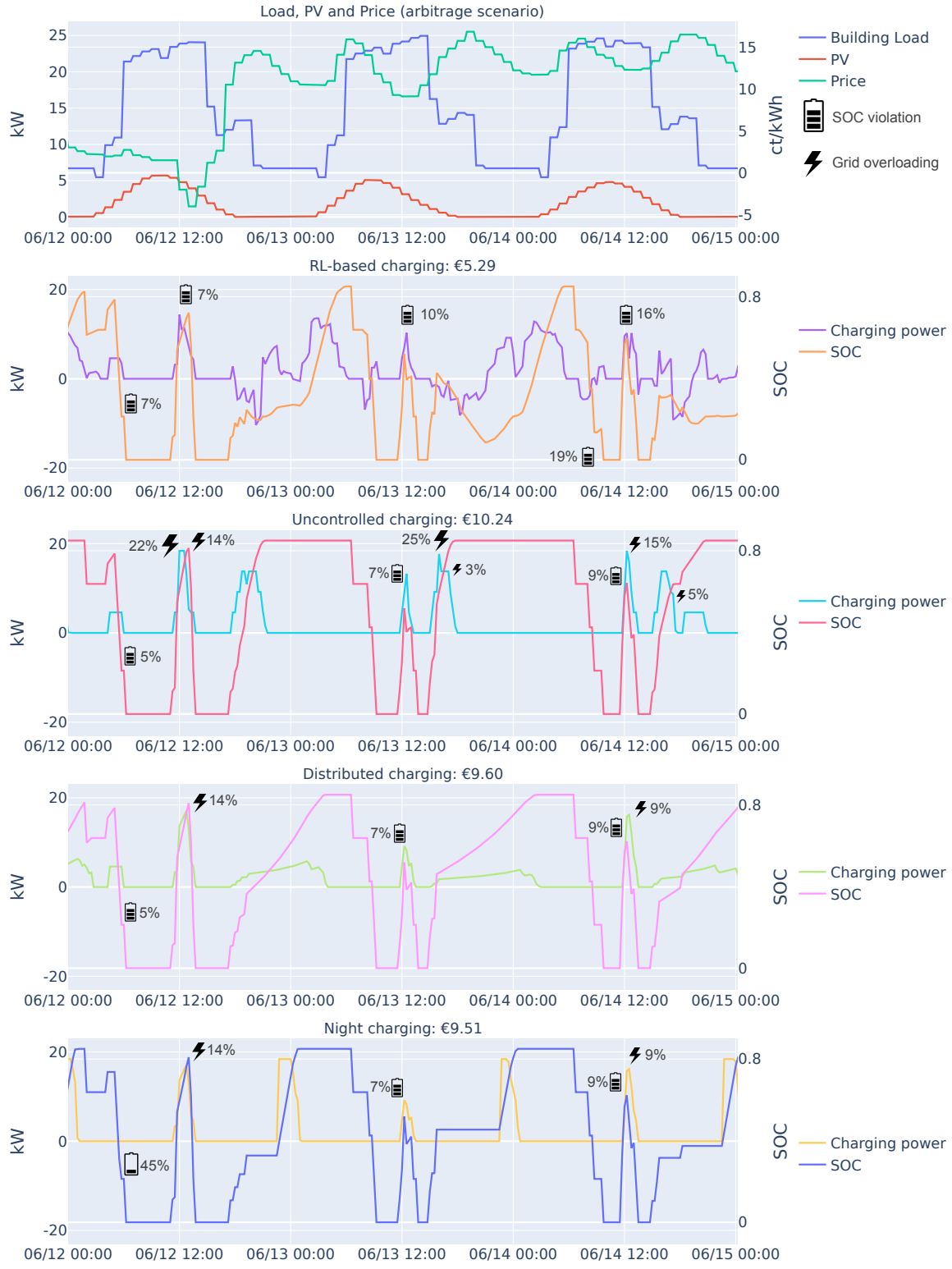


Figure 6.1.3: 5 EV agents and static benchmarks [1]

[1]The battery symbol in Figure 6.1.3 signals how full the battery was upon departure - the percentage signals how much SOC was missing when the vehicle departed. The prices in the subplot titles refer to the total spending during the displayed timeframe. Wherever an SOC increase is observed without a positive charging power, vehicles arrive in different time steps with different SOC levels, thereby increasing the SOC curve. A vice versa effect can be seen when vehicles depart the station.

The RL agent discharges the battery on arrival, exploiting the high spot market prices. When prices are low, the energy is charged back just in time before the vehicles leave for the next trip. This results in overall savings of around 50% compared to the shown benchmarks.

A higher level of SOC violations can be observed for RL, although all strategies violate the SOC requirement to some extent in the tightly constrained caretaker use-case. Note however how no grid overloadings took place with the RL agent: A trade-off was therefore made between slightly higher SOC violations and avoiding grid overloadings. This trade-off would probably be taken in the real-world as well, considering that the consequences of a grid overloading could be a blown fuse and a halt of charging for all vehicles.

The night charging strategy shows a severe SOC violation, because a vehicle left for an emergency trip and only returned at 4:00 am. The charging strategy was thereby not able to adjust to this, resulting in a large amount of missing energy. On Jun 13 at 16:00, the dumb charging strategy overloads the grid connection severely because all 5 cars start charging upon arrival while there is a lack in PV and the building load has not decreased sufficiently. The RL agent and the other static benchmarks avoid this violation. Combining the insights of Figure 6.1.3 with Figure 6.1.2, it can be seen that RL-based charging is the most effective in avoiding overloading violations out of all charging strategies.

While the 5 EV agent managed to learn an effective charging strategy, it was found that the economic results did not scale linearly with the number of vehicles. Having trained both agents for approximately the same number of time steps, it was found that the economic performance of the 5 EV agent was worse due to the added complexity of the problem. Figure 6.1.1 suggests that the training process was not concluded at 5.5M steps, because the training reward curves had not plateaued at the end of the experiment. It is therefore likely that performance would have increased further with a longer training time, although progress might have been slow given the shallow slope of the reward curve.

Further analyses are therefore conducted on the fully trained agent with 1 EV, because it is the least likely agent to have been impacted by a constrained computational budget. The next sections will thus closely analyse the charging strategy of the 1 EV agent. To present a final business case for the originally sized delivery-, caretaker- and utility

company, the economic results of the 1 EV agent are scaled up to match the number of vehicles defined for each use-case (50, 30, and 25, respectively). This is done by multiplying the expenses by the respective number of vehicles. Naturally, this poses a potential limitation. It should be assessed in future work whether the same economic performance can be reached when 50 vehicles are trained simultaneously, and a sufficient amount of computational resources are used. The full results of the 5 EV agent are listed in Appendix B for reference.

6.2 Use-case analysis

This section has a closer look at the three commercial use-cases, their charging strategies and their economic results. RL is compared to its benchmarks and important differences in performance or applicability are highlighted. A final comparison between the use-cases and the charging strategies is conducted in the following section.

6.2.1 Last-mile delivery

The charging strategies of the arbitrage and tariff use-case are shown in Figure 6.2.1 for RL, Linear Programming (LP), and Uncontrolled Charging (UC). In the arbitrage scenario, the charging strategy of RL aligned well with the linear optimization result. As originally assumed, the agent recognized the intraday price differences, and discharged at peak times to buy back the energy later on at lower prices. In the tariff scenario, a small portion of discharging remained in the tariff scenario. This was either due to negative electricity prices at night, or due to sub-optimal decision-making in some cases. Since the agent still received an immediate positive reward for discharging (PV feed-in tariff), and only encountered the disadvantages in the future (more expensive spot-price), a trade-off was likely made between immediate and future rewards. This could potentially be mitigated by setting the discount factor to $\gamma = 1$. The linear optimization rarely decided to discharge upon arrival. In these cases, discharging was advantageous due to negative prices at night. LP therefore discharged upon arrival to have more kWh to charge at negative prices. Overall, RL matches the behaviour of LP well, even though it was only given real-time available information instead of 1 year of data.

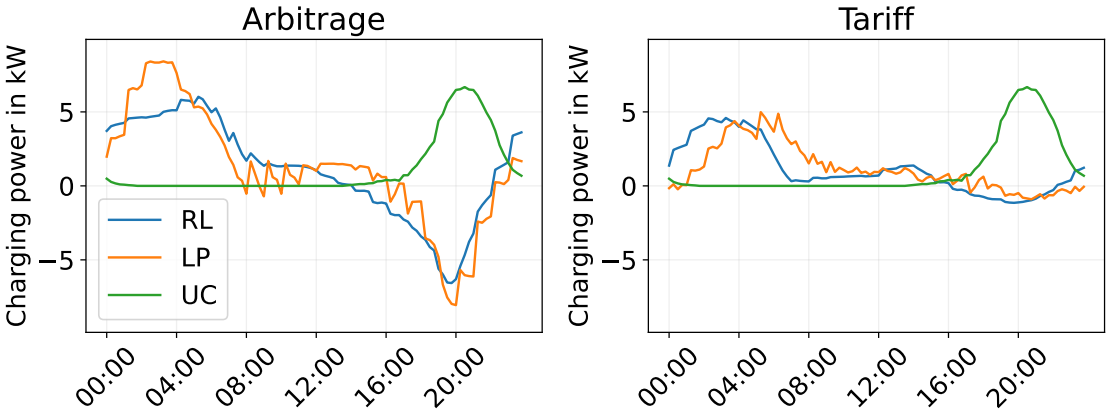


Figure 6.2.1: Last-mile delivery - Charging strategy for RL, LP and dumb charging

Figure 6.2.2 visualizes the charging strategy from the action perspective. While the RL agent smoothly distributed its actions across the available range, UC and LP stick to the edge cases. As mentioned above, a small penalty was given to the agent if it sent a charging signal to an empty spot. Judging from the distribution, the agent seems to have learnt to avoid invalid actions.

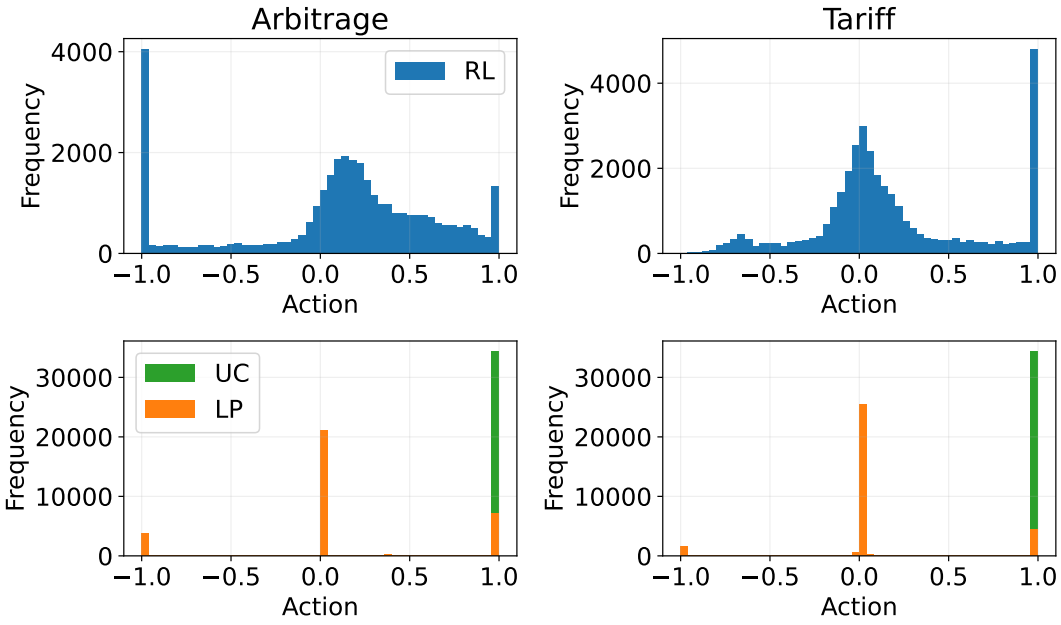
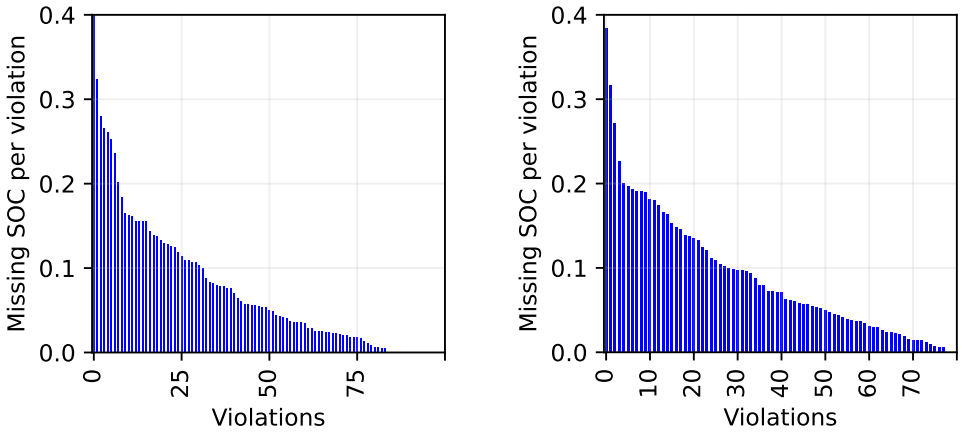


Figure 6.2.2: Last-mile delivery - Action distribution

Figure 6.2.3 shows the number and severity of SOC violations for one year of operations. A violation occurs when a car departs with less than the target SOC.



(a) Arbitrage (b) Tariff

Figure 6.2.3: Last-mile delivery - SOC violations for RL-based charging

In the arbitrage and tariff scenario, 25 and 30 violations exceeded 10%, respectively. These occur less than 1% of the year, but potentially represent unacceptable violations, as operations might be endangered due to insufficient energy. In total, the 82 and 78 violations yield 7.5 and 7.3 missing batteries over the entire year. Dumb charging did not violate the target SOC requirements. No grid connection overloading took place when running the model with only 1 car at a time. The linear optimization did not violate SOC constraints in any scenario.

Figure 6.2.4 compares battery degradation across the charging strategies. A difference of ca. 1% can be observed. RL thereby managed to achieve the lowest degradation, and the linear optimization strategy degraded the battery the most. This can be partly attributed to the SOC violations of RL, which amounted to 7 batteries over the year. However, since rainflow cycle counting was used, the severity and depth of the cycles were taken into account. Note how in Figure 6.2.1 the linear optimization profile shows pronounced spikes which cannot be observed in the other two strategies. These spikes might have contributed to the increased battery degradation. In the tariff scenario, the degradation difference was smaller, although RL still achieved the lowest degradation.

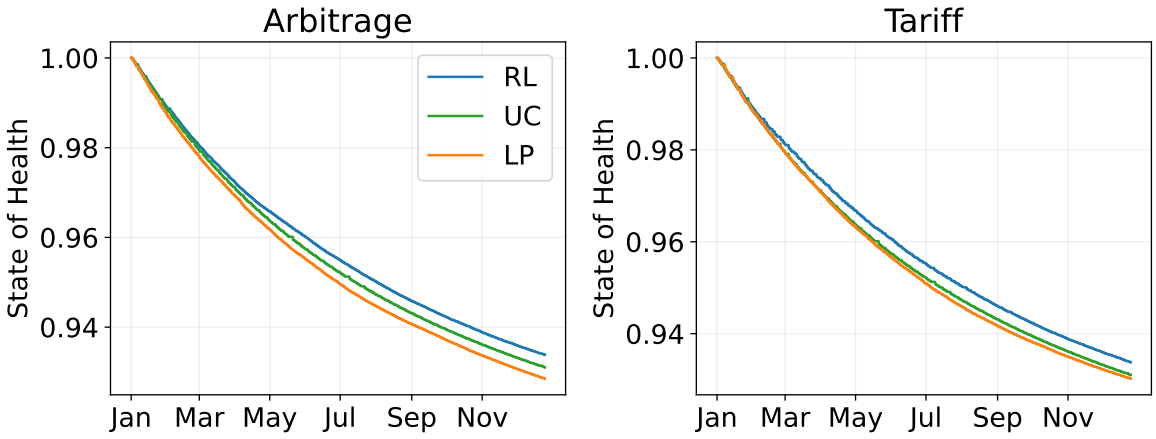


Figure 6.2.4: Last-mile delivery - SOH for RL, LP and dumb charging

Over the battery's lifetime, the difference becomes apparent: with RL-based charging, the battery would last ca. 1 year longer in the arbitrage scenario, as shown in Figure 6.2.5. Premature battery degradation will be further assessed economically by taking into account an earlier investment in a new battery.

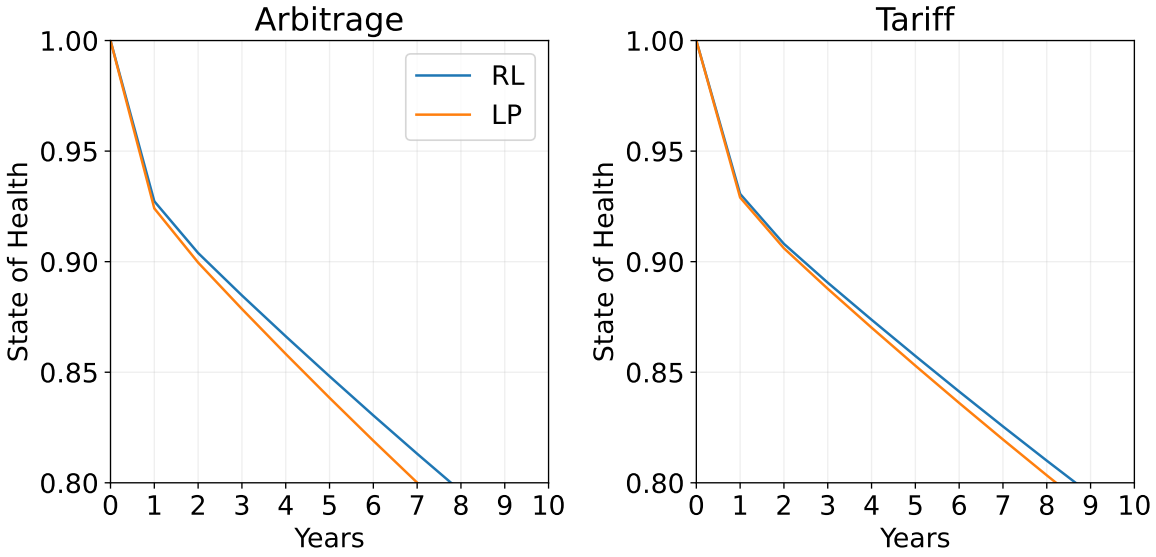


Figure 6.2.5: Last-mile delivery - Degradation comparison over the battery lifetime

Having a closer look at the economic evaluation, it becomes evident that considerable savings can be achieved across all static benchmarks with RL-based charging. Figure 6.2.6 shows the absolute spendings on electricity for each strategy, as well as the relative savings compared to RL-based charging.

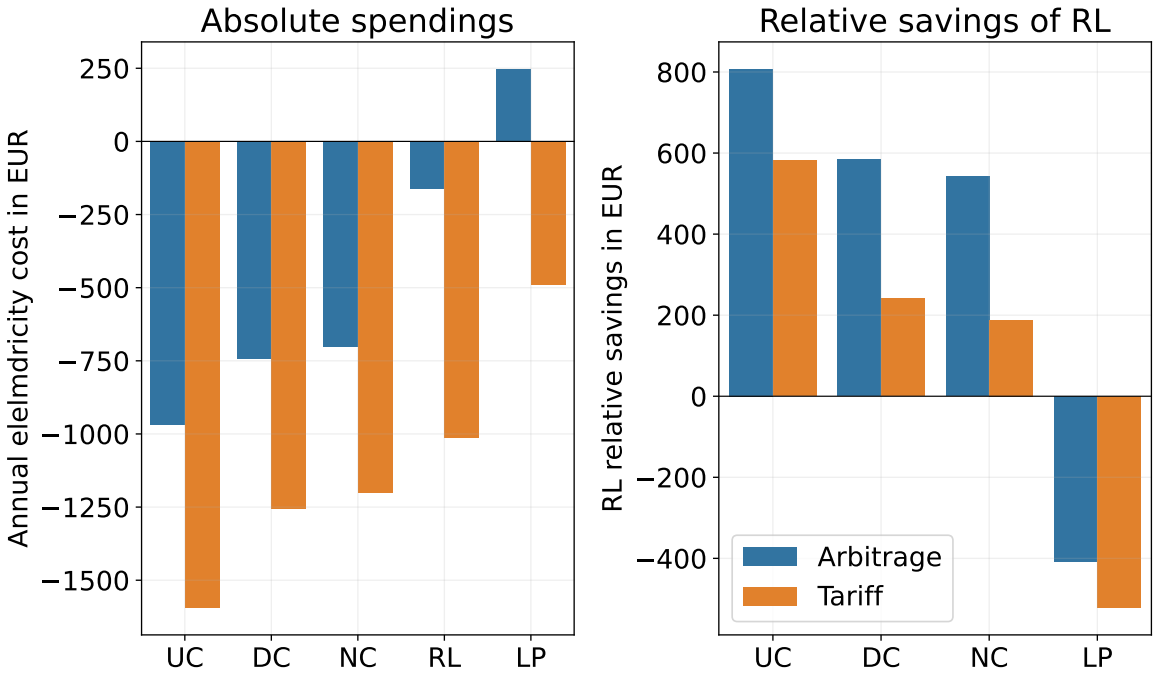


Figure 6.2.6: Last-mile delivery - Spendings and relative savings

On the x-axis, the charging strategies are shown: Uncontrolled Charging (UC), Distributed Charging (DC), Night Charging (NC), RL, and LP. Where necessary, results have been corrected for missing energy with the average spot price of 2021. As expected, linear programming with perfect knowledge outperformed RL, which only possessed real-time information.

Figure 6.2.7 shows the discounted savings for the last-mile delivery use-case with 50 vehicles and a load controller of €1200. The values at year 10 thereby represent the Net Present Value (NPV) of the investment decision (switching RL-based charging). The point at which savings become positive is the break-even point of the investment decision.

Apart from the controller, initial investments included an upgrade to a smart-charging-enabled EV charger (4.6kW - 22kW: €300-€500). Further, 10 % were deducted from the savings to take into account third party fees (e.g. balancing service provider or spot market broker). A 7.5% discount rate was assumed for the calculations. Battery replacement costs were assumed at €15000, €10000 and €7500 for 60 kWh, 50 kWh, and 16.7 kWh batteries, respectively. In the arbitrage use-case, an economically feasible result is reached within two to three years for all use-cases and across all static benchmarks. In the tariff scenario, a switch from distributed and night charging only

becomes economically feasible after 5 years.

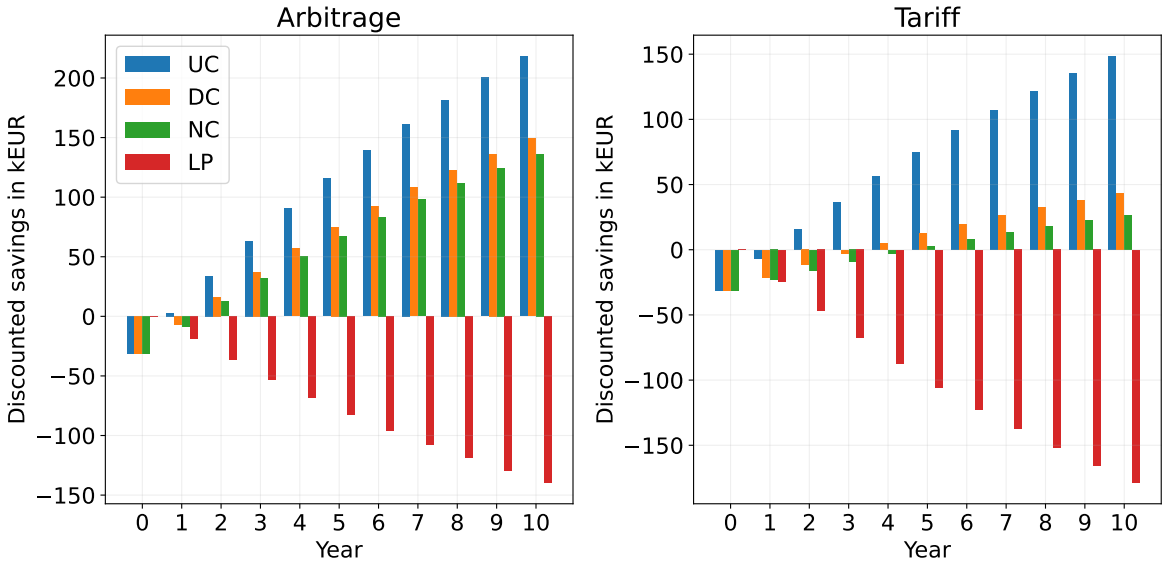


Figure 6.2.7: Last-mile delivery - Discounted savings [2]

6.2.2 Caretaker

Figure 6.2.8 visualizes the charging strategies for the caretaker use-case. The caretaker use-case featured possible emergency trips at night (2am-4am) with a 2% probability, as well as short lunch breaks which were not always long enough to charge until the target SOC. Under these conditions, LP did not find a feasible solution. The constraints of the optimization had to be relaxed manually for the lunch break and between 2am and 4am until a feasible solution was found. This was only possible after considerable reductions of the target SOC. It was also tested to incorporate the constraints of target SOC and grid overloading in the objective function instead of posing hard constraints - this would have led the LP algorithm to optimize a reward function similar to RL. However, this method did not lead to meaningful results within the allocated time frame for implementing the LP model, and is therefore not discussed in this study. As a result, the LP-based charging strategy shows insufficient charging during the lunch breaks. RL, on the other hand, aligns well with uncontrolled charging during the lunch breaks and leverages the intraday price differences throughout the night, similar to the LP strategy.

[2]Note how the LP strategy does not have an initial investment. This is because LP would require the same technical components - a switch from LP to RL would therefore not require initial investments.

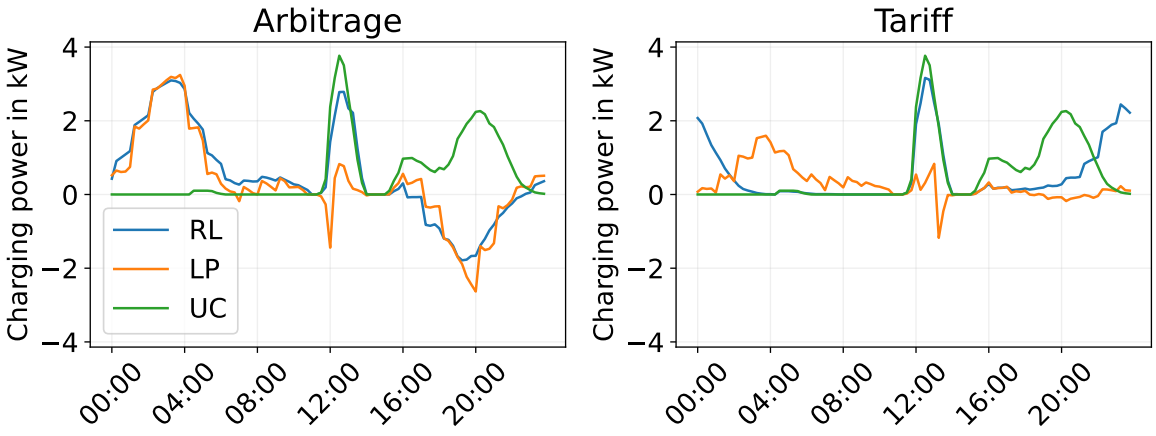


Figure 6.2.8: Caretaker - Charging strategy

Figure 6.2.9 illustrates the action distributions of the three strategies. Again, the RL agent distributes its actions more evenly between -1 and 1, whereas dumb and LP charging stick to zero and the edge cases. The peak at 1 for the RL agent is due to the lunch break, where maximum power is often required to satisfy target SOC constraints.

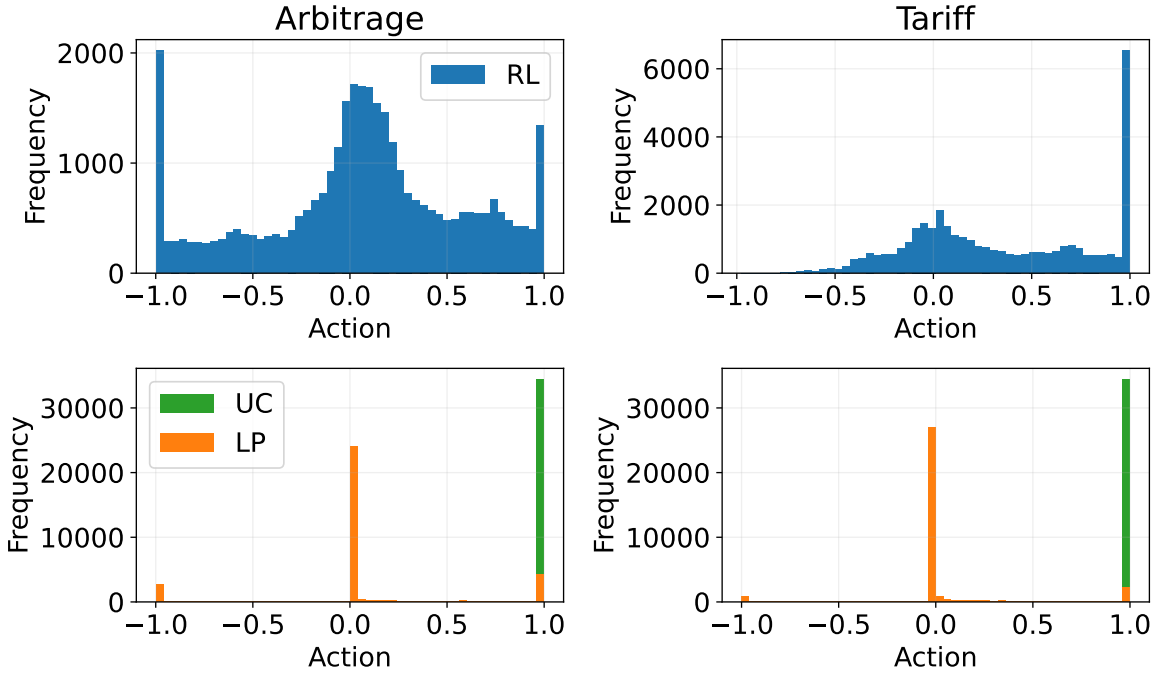
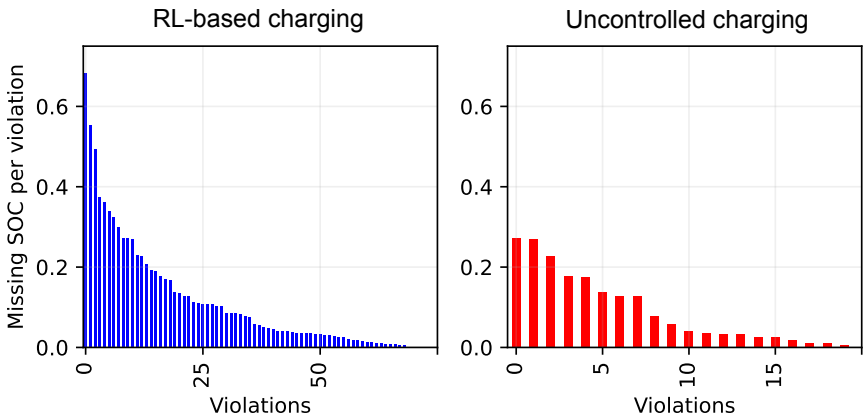


Figure 6.2.9: Caretaker - Action distribution

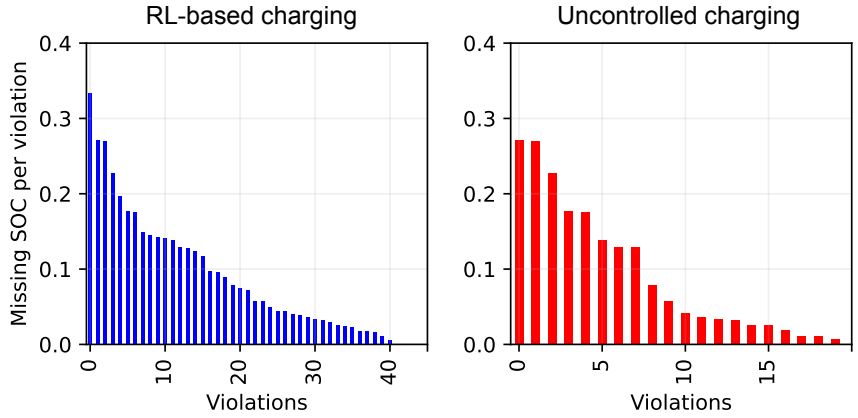
Figure 6.2.10 illustrates the SOC violations. As mentioned, not all lunch breaks were long enough to reach the target SOC. Therefore, SOC violations even occur with the

dumb charging strategy: 20 violations with 2 missing batteries occur in total for both the arbitrage and tariff scenario. For RL-based charging, 69 violations with 9 missing batteries, and 41 violations with 4 missing batteries occur for the arbitrage and tariff scenario, respectively.

Due to the manual constraint relaxation, LP-based charging violated the SOC constraint every day during the lunch break, and sometimes at night due to an emergency trip. For each use-case, 375 violations with 94 missing batteries were found. It becomes evident that the chosen LP approach with relaxed constraints yielded an impractical solution. While this presents a potential limitation for LP-based approaches in tightly constrained mobility problems, it needs to be said that the full spectrum of LP solutions was not exhausted. Future work could therefore include Model-Predictive Control (MPC) implementations, a different objective function, or a revision of the chosen approach, potentially achieving a better performing LP model.



(a) Arbitrage



(b) Tariff

Figure 6.2.10: Caretaker - SOC violations for RL and UC

Figure 6.2.11 shows the progression of SOH for the caretaker use-case. Note that LP shows lower degradation due to the numerous SOC violations.

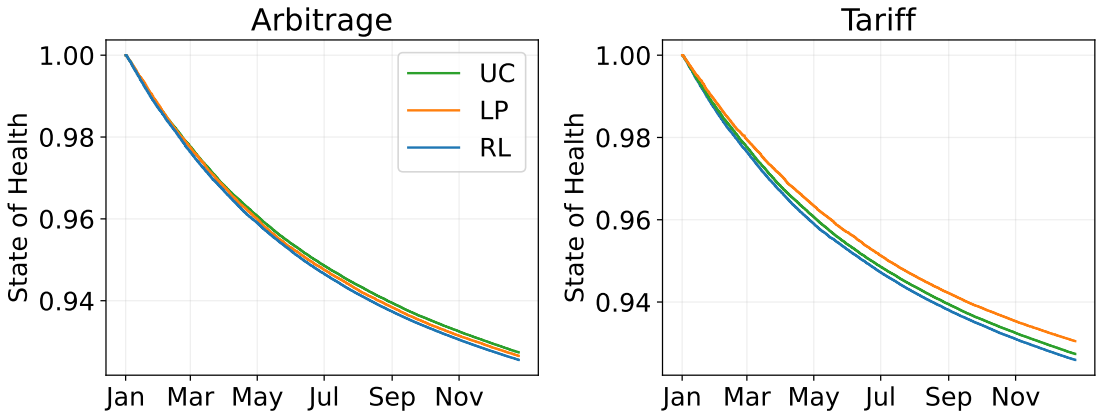


Figure 6.2.11: Caretaker - SOH for RL, LP and dumb charging

Over a time horizon of 10 years, the battery would have lasted 1 or 1.8 years longer, for the arbitrage and tariff scenario, respectively. Naturally, these results do not hold significant value, since the mobility constraints were violated to such an unacceptable extent that operation would never be maintained at this level for 10 years.

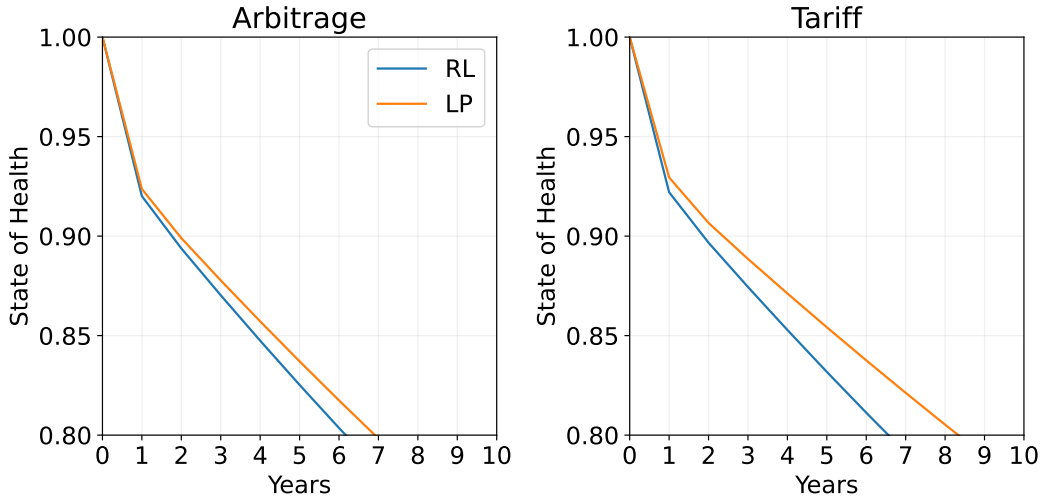


Figure 6.2.12: Caretaker - Degradation comparison over the battery lifetime

The economic results for the caretaker use-case are shown in Figure 6.2.13. Missing energy is corrected for with the average spot prices of 2021. As can be seen, RL-based charging outperforms every static charging strategy, both in the arbitrage and in the tariff scenario.

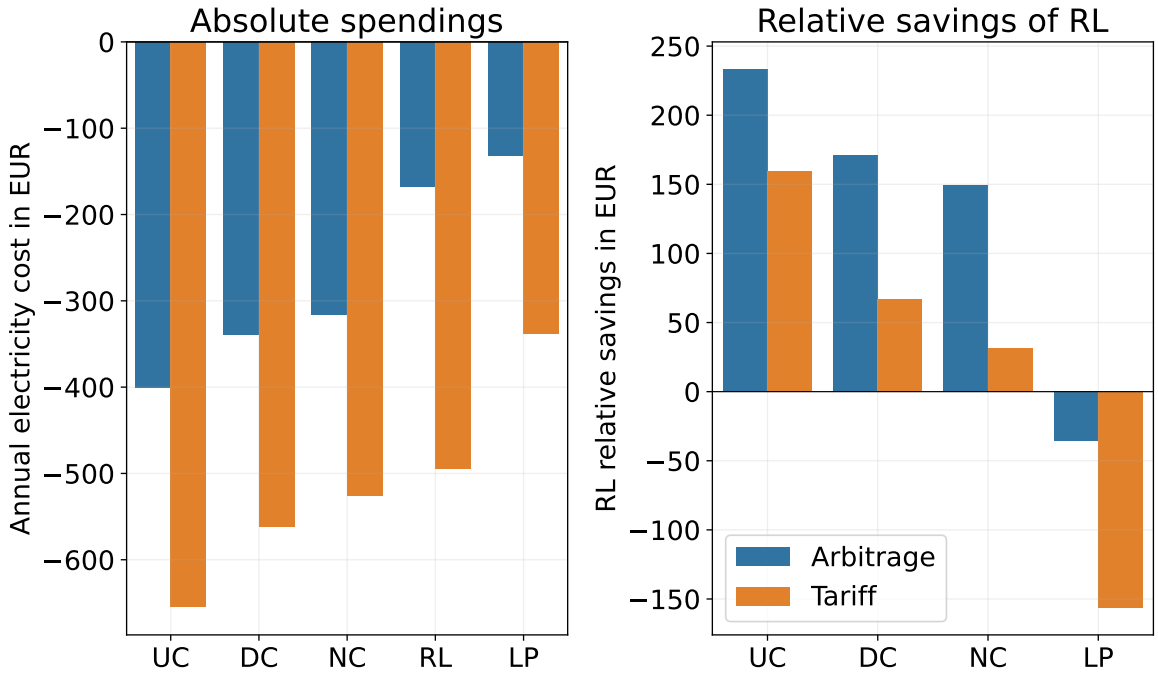


Figure 6.2.13: Caretaker - Spendings and relative savings

Note how the savings in the tariff scenario are considerably smaller due to the fact that energy arbitrage is no longer an economically feasible option: There are fewer degrees of freedom to generate savings in the tariff scenario. LP is not outperformed, even when missing electricity is bought back at average spot prices, suggesting that perfect knowledge for the entire year grants a significant advantage.

Figure 6.2.14 shows the discounted savings for the caretaker use-case with 30 vehicles. In the arbitrage use-case, an economically advantageous outcome is reached for all static benchmarks. Apart from dumb charging, the savings are not sufficiently large to displace the investments within ten years in the tariff scenario. In this case, a switch to RL-based charging would only make sense if the necessary chargers were already installed. This result confirms observations from literature and recent market developments: it might be necessary to tap into multiple revenue streams when setting up a charging management business model. Such revenue streams could include energy arbitrage, as shown here, vehicle-to-grid, vehicle-to-building, or other balancing services.

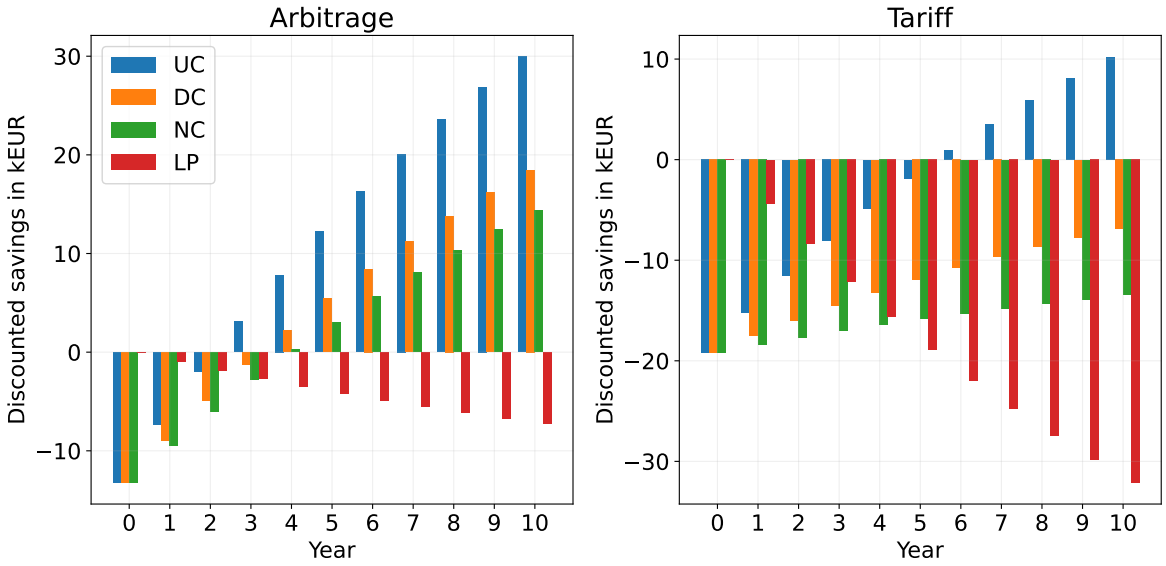


Figure 6.2.14: Caretaker - Discounted savings

6.2.3 Utility

Figure 6.2.15 shows the charging strategies of UC, LP, and RL charging. RL aligns well with LP in both scenarios. Similar to last-mile delivery, some discharging can be observed upon arrival in the tariff scenario; this could be potentially improved by setting the discount factor to $\gamma = 1$.

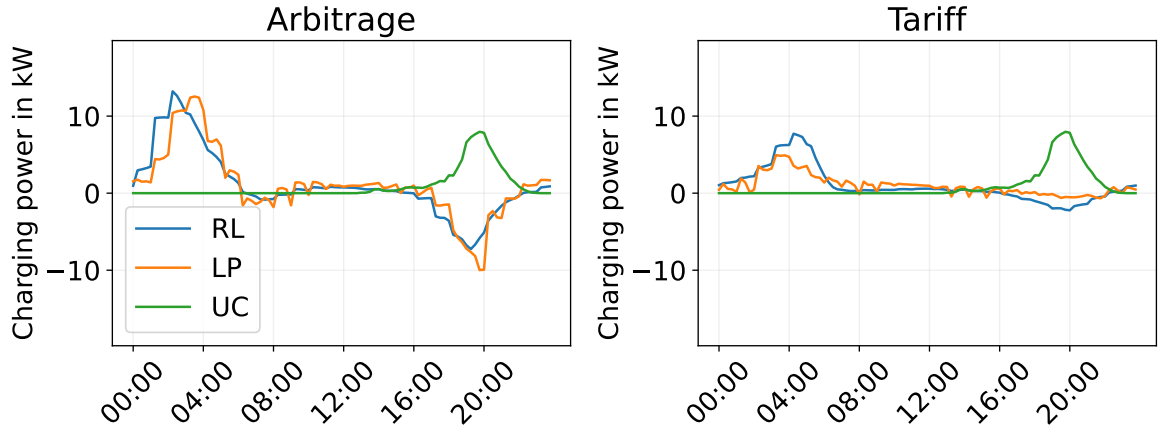


Figure 6.2.15: Utility - Charging strategy

Figure 6.2.16 shows the action distribution for the utility. Actions are evenly spread out, and a distribution around zero can be observed. UC and LP stick to the edge cases.

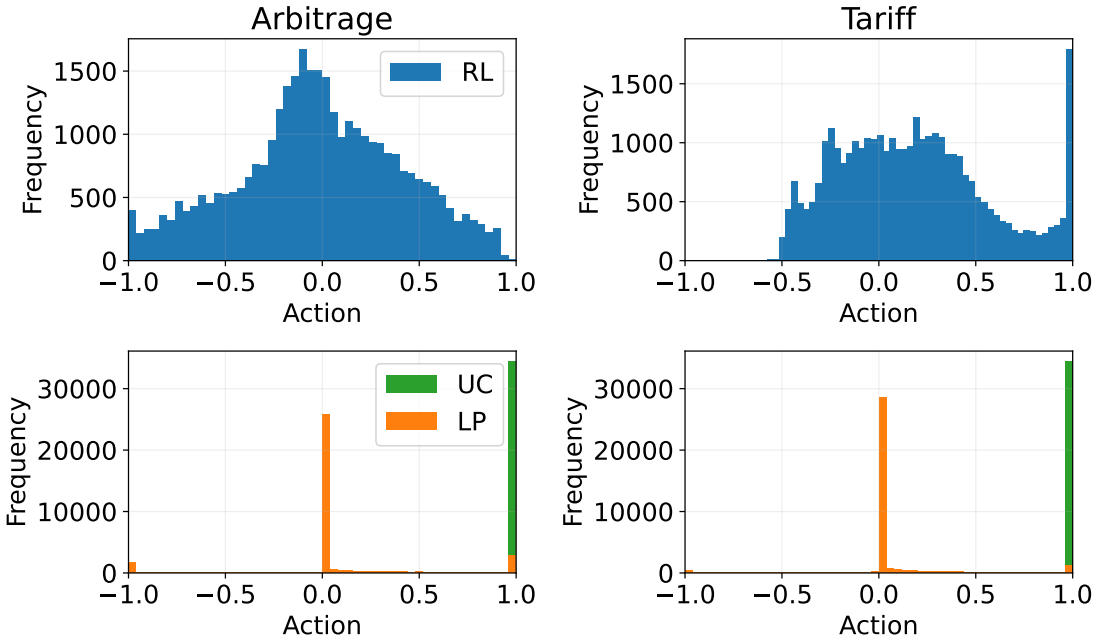
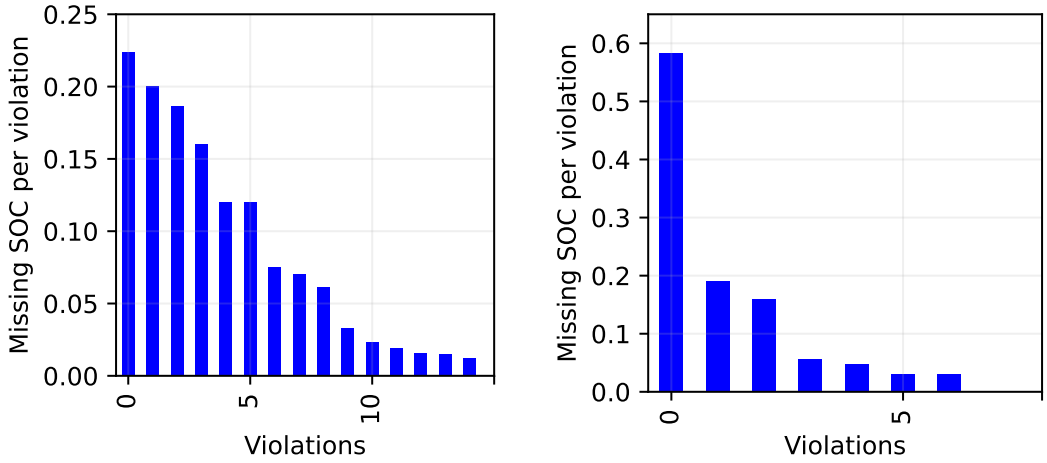


Figure 6.2.16: Utility - Action distribution

In the utility use-case, the number of SOC violations was particularly low, as shown in Figure 6.2.17. Over the entire year, 14 violations took place in the arbitrage scenario, with an equivalent of 1.5 batteries. In the tariff scenario, 6 violations took place with an equivalent of 1 battery. Note that one violation was ca. 58% and would have represented an unacceptable violation, potentially rendering the car unusable for the upcoming trip. The LP charging strategy did not show any violations.



(a) Arbitrage

(b) Tariff

Figure 6.2.17: Utility - SOC violations for RL-based charging

Figure 6.2.18 shows the battery degradation over 1 year of operations. In both scenarios, a difference of less than 0.5% can be observed between the strategies.

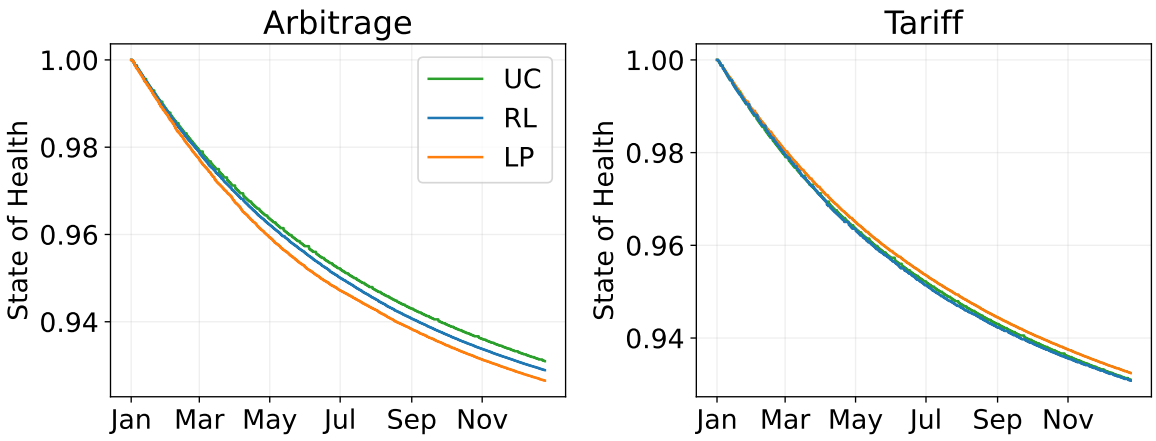


Figure 6.2.18: Utility - SOH for RL, LP and dumb charging

A similar result was observed over the course of 10 years. In the arbitrage scenario, a new battery was due ca. 6 months earlier with the LP strategy than with the RL-based strategy, and vice versa for the tariff scenario.

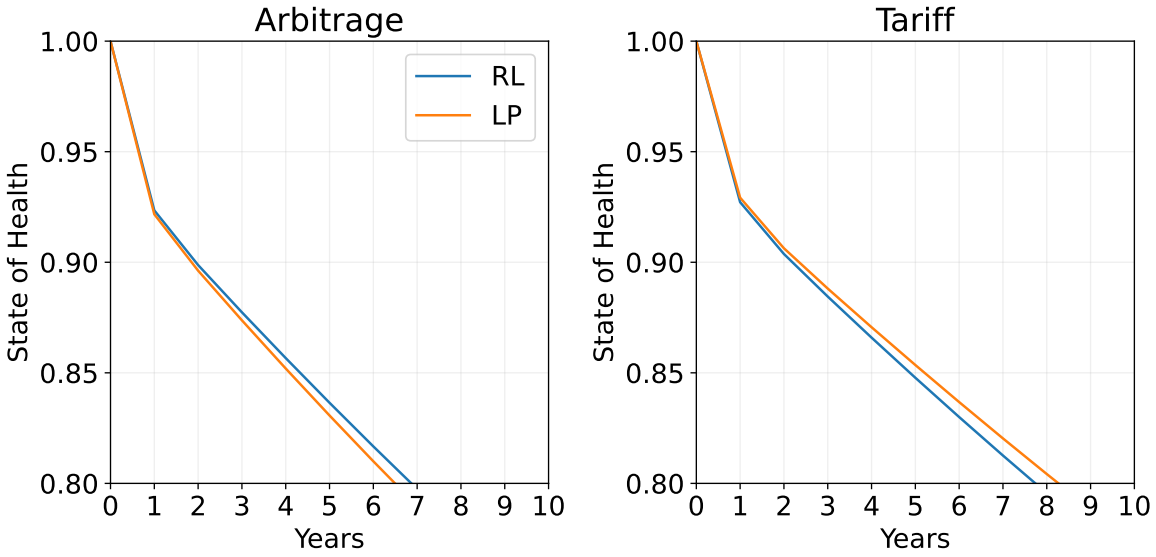


Figure 6.2.19: Utility - Degradation comparison over the battery lifetime

Looking at the economic results, it was found that RL was not able to outperform the distributed and night charging strategy in the tariff scenario, as shown in Figure 6.2.20. This suggests that the utility use-case features some additional difficulties that could

arise from the intricate combinations of battery capacity, vehicle schedules, building load, PV feed-in, and charger power.

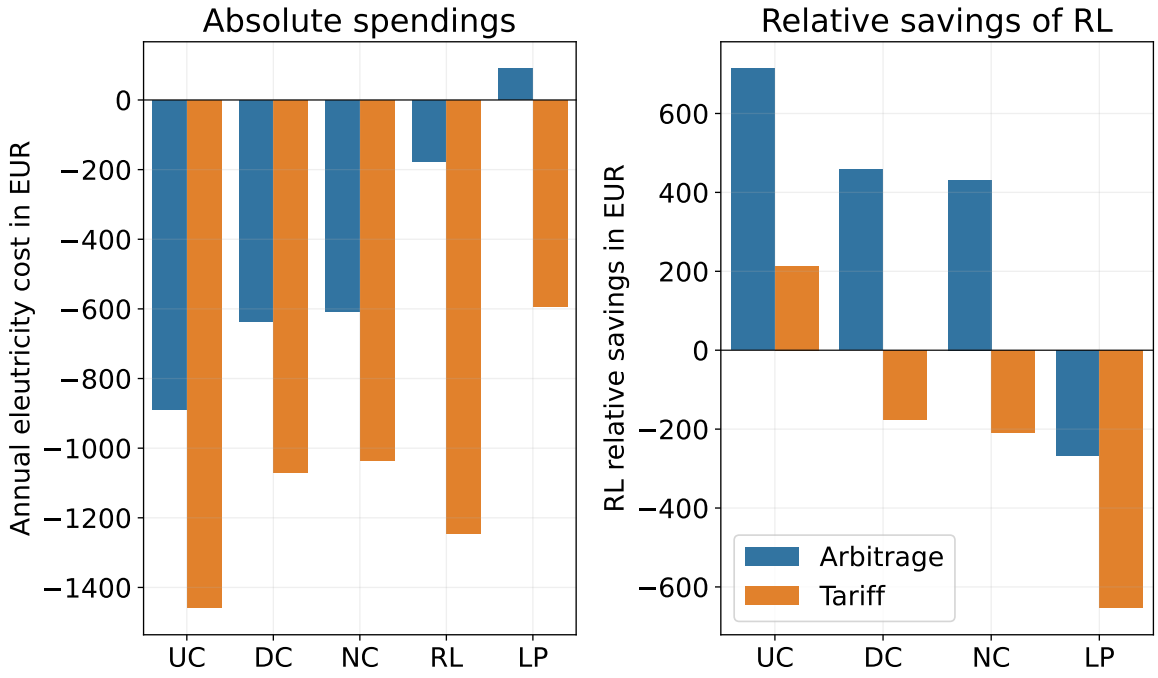


Figure 6.2.20: Utility - Spendings and relative savings

Figure 6.2.21 presents the discounted savings for the utility use-case with 25 vehicles. In the arbitrage scenario, a positive result is reached for all strategies.

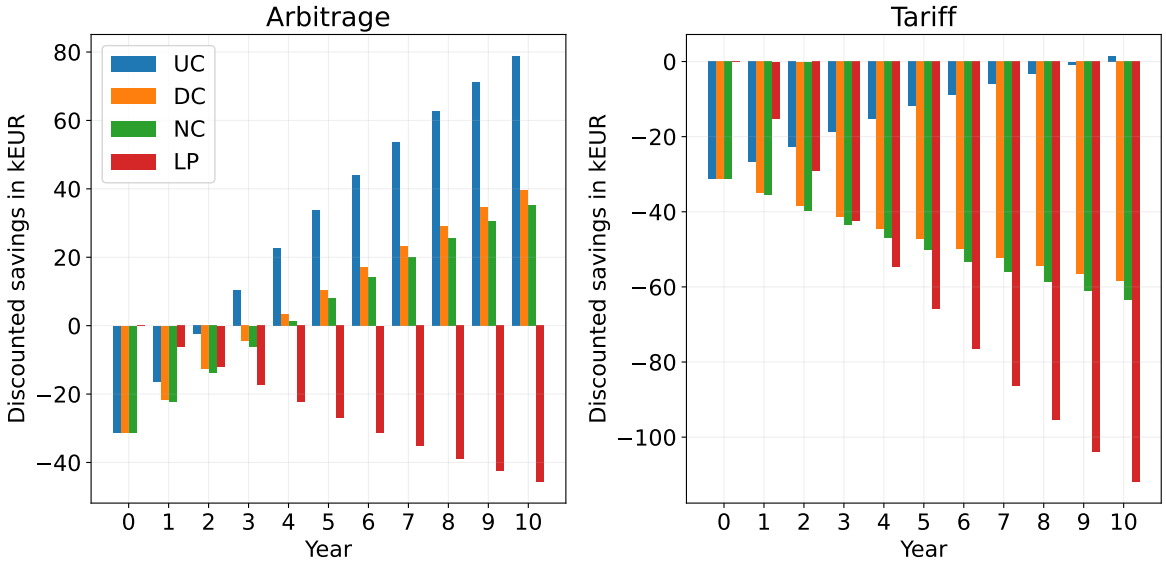


Figure 6.2.21: Utility - Discounted savings

In the tariff scenario, only the switch from uncontrolled charging yielded a break-even after 10 years. Given the infrastructure and expertise, the utility company is the most likely to enable additional revenue streams, such as arbitrage, vehicle-to-grid, or other balancing approaches. An economically beneficial result is therefore likely for the utility company.

6.3 Comparative analysis

Tables 6.3.1, 6.3.2, and 6.3.3 enable a final performance comparison between RL and its benchmarks. The SOC fulfilment of RL stayed above 95% for all use-cases and scenarios. For CT, manual adjustments had to be made to the LP problem, until a feasible solution was found. At that point, the LP strategy resulted in an impractical solution with ca. 68% fulfilment. Degradation stayed relatively similar throughout all cases.

LMD arbitrage						**LMD tariff**				
	RL	LP	UC	DC	NC	RL	LP	UC	DC	NC
Cost in EUR/a	160.9	-247.9	968.6	744.9	702.9	1,086.1	490.9	1,594.8	1,254.7	1,200.8
kWh charged	8977.4	9427.4	9427.4	9427.4	9427.4	8,989.4	9,427.4	9,427.4	9,427.4	9,427.4
kWh violated	450	0	0	0	0	438.0	0.0	0	0	0
Avg ct/kWh	1.71	-2.63	10.27	7.90	7.46	11.52	5.21	16.92	13.31	12.74
SOC % fulfilled	95.23%	100.00%	100.00%	100.00%	100.00%	95.35%	100.00%	100.00%	100.00%	100.00%
Degradation	6.62%	7.15%	6.90%	6.81%	6.90%	6.62%	6.83%	6.90%	6.81%	6.90%

Table 6.3.1: Summary LMD

CT arbitrage						**CT tariff**				
	RL	LP	UC	DC	NC	RL	LP	UC	DC	NC
Cost in EUR/a	167.3	131.6	400.8	338.7	316.3	494.5	338.0	654.4	561.2	525.7
kWh charged	4,720.8	3,309.8	4,862.9	4,862.9	4,783.1	4,796.4	3,276.4	4,862.9	4,862.9	4,783.1
kWh violated	142.1	1,553.1	0	0	79.8	66.5	1,586.5	0	0	79.8
Avg ct/kWh	3.44	2.71	8.24	6.96	6.50	10.17	6.95	13.46	11.54	10.81
SOC % fulfilled	97.08%	68.06%	100.00%	100.00%	98.36%	98.63%	67.38%	100.00%	100.00%	98.36%
Degradation	7.45%	6.99%	7.26%	7.24%	7.23%	7.41%	6.68%	7.26%	7.25%	7.23%

Table 6.3.2: Summary CT

UT arbitrage						**UT tariff**				
	RL	LP	UC	DC	NC	RL	LP	UC	DC	NC
Cost in EUR/a	176.6	-89.7	890.0	635.2	607.5	1,245.5	592.6	1,458.5	1,069.4	1,036.2
kWh charged	7,994.8	8,061.3	8,061.3	8,061.3	8,061.3	8,006.5	8,061.3	8,061.3	8,061.3	8,061.3
kWh violated	66.5	0.0	0	0	0	54.8	0.0	0	0	0
Avg ct/kWh	2.19	-1.11	11.04	7.88	7.54	15.45	7.35	18.09	13.27	12.85
SOC % fulfilled	99.18%	100.00%	100.00%	100.00%	100.00%	99.32%	100.00%	100.00%	100.00%	100.00%
Degradation	7.11%	7.35%	6.90%	6.81%	6.91%	6.75%	6.84%	6.90%	6.81%	6.90%

Table 6.3.3: Summary UT

Tying the results together, Figure 6.3.1 illustrates the different savings potentials that RL could bring to the use-cases. LMD seems to yield the highest savings per car due to the large battery of 60 kWh, and the regular vehicle schedules with long durations of stay. Comparing RL with UC in the LMD use-case, a cost reduction of 83.4% can be observed. This aligns well with savings reported in literature, which amount to 80% [8]. Comparing the savings of LP vs. UC and RL vs. UC, it can be seen that the savings potential of RL is ca. 35% lower for the LMD use-case. This also aligns well with findings in the literature, where the savings potential was found to be ca. 40% lower [113]. UT closely follows with its 50 kWh batteries, 22 kW chargers and its similar schedules: up to 80% savings can be observed compared to UC. CT has the lowest annual savings per car due to its smaller battery of 16.7 kWh. Its maximum percentage savings amount to 58%.

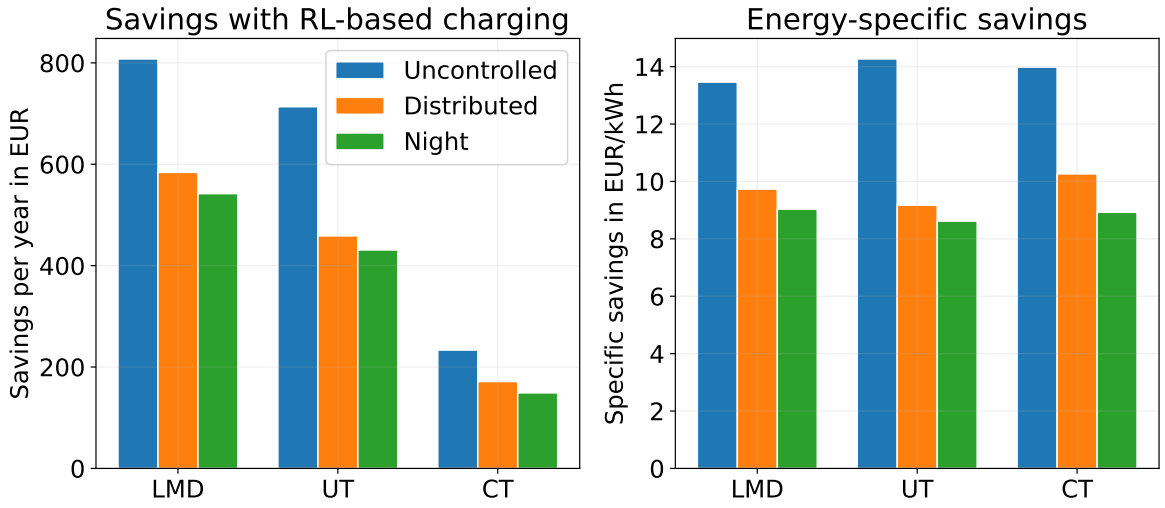


Figure 6.3.1: Savings comparison of RL vs. static benchmarks

However, if we look at the graph on the right, we can see that the savings potential is very similar when adjusting for the different battery sizes: The RL agent secured almost identical savings per kWh for all three use-cases. Since building load, PV, or EV schedules did not have a negative impact on the specific savings, it can be concluded that all three commercial use-cases are well-suited for RL-based EV charging optimization.

6.4 Implications on sustainability

The results of this study indicate that RL is well-able to learn how to solve complex problems with uncertainty and real-time information. By optimizing energy consumption patterns, including electric vehicle charging, heating, cooling, and other essential services, RL can play a vital role in creating intelligent and responsive urban ecosystems. This aligns with SDG 11 (Sustainable Cities and Communities), promoting resilient and efficient urban environments that adapt to real-time demands. Within the broader energy landscape, RL's application extends to grid management, renewable energy integration, and demand-side management, contributing to SDG 7 (Affordable and Clean Energy) and SDG 9 (Industry, Innovation, and Infrastructure). Furthermore, by paving the way for increased renewable energy penetration, SDG 13 (Climate Action) is directly supported. From individual households to large-scale industrial complexes, the implementation of RL-based energy management systems can facilitate a more sustainable, and economically feasible energy future. Its role in unlocking flexibility, reducing emissions, and enhancing grid stability positions RL as a potential cornerstone technology of the energy transition. In considering the core objectives of sustainable development, RL's role in the energy landscape resonates with both environmental integrity and equity across generations. By optimizing energy consumption and enhancing renewable energy integration, RL sustains natural resources, promoting an equilibrium between the environment and human needs.

It is important to mention here the trade-off between AI-capabilities and deterministic, explainable solutions. Critical areas such as the transmission grid still rely on simple, deterministic models for forecasting, because they are bound by regulation to be explainable. Research and testing are needed to explore where AI-based solutions such as Deep Reinforcement Learning can be leveraged in a safe, and reliable manner. Only once this is clear can a broad positive impact be made on the energy transition.

6.5 Discussion

RL managed to learn an effective charging strategy, both for 1 EV and 5 EVs. Trying to mimic real-world conditions, the agent only received real-time information and did not possess information about upcoming arrivals. Overall, SOC violations stayed within an

acceptable range and grid violations were avoided effectively. Three static benchmarks and a linear optimization were introduced to compare RL to commonly implemented methods. RL outperformed the static benchmarks. Linear optimization with perfect knowledge provided an indication of what a potential best-case scenario could look like. However, LP-based charging also showed practical limitations: it is unrealistic to know one year of data ahead of time. Further, LP did not find a feasible solution in the tightly constrained caretaker use-case. A feasible solution was only found after significant constraint relaxation, leading to a practically unacceptable solution. From the rule-based methods, distributed charging seemed to perform particularly well, finding a trade-off between leveraging night prices and not overloading the grid with a sudden increase in charging power. However, one would have to re-program the vehicle schedule every day, based on the most recent arrival and departure time. Further, distributed charging requires a flexible adjustment of charging power based on the duration of stay - a feature that is not commonly implemented in vehicles, and especially rare in non-smart-charging EV chargers. Regarding battery degradation, it was found that RL did not put large additional strains on the battery. On the contrary, RL was found to be the most preserving charging strategy in some cases - even when engaging in arbitrage.

A positive business case could always be made for switching to RL-based charging with arbitrage. Without arbitrage, the switch to RL in the utility use-case was only feasible when switching from uncontrolled charging. However, it is a realistic assumption that utilities are early adopters in terms of additional revenues streams due to arbitrage, V2H, V2G or other balancing features. This led to the conclusion that an economically feasible business case can be found for all three commercial use-cases. Looking at sustainability and implications for the energy transition, RL-based energy management systems show promising potential to be able to adapt to the rapidly changing energy landscape. The results of this study indicate that the versatility and performance of RL seem a good match for unlocking flexibility potential in different commercial use-cases. Going hand in hand with transmission operators, energy management solutions such as this one could enable an increasing penetration of renewables in the grid.

Nevertheless, limitations can be found within the methodology and implementation. First, training RL agents has proven to be a computationally intensive, and time-consuming task. While the training of the final set of agents ”only” took around

48 hours on 1 Nvidia GTX 3090 GPU and 28 Intel i9 CPUs (ca. 40 Teraflops), countless attempts and computations went into getting there. Adding up extensive hyperparameter studies, testing, trial and error, and iteratively improving the software implementation, yields computational times in the order of weeks, with workloads distributed across multiple remote workers. Naturally, this could have been sped up by making High-Performance Computing (HPC) resources available. This, however, is always a trade-off between resource requirements and potential research outcomes. Having successfully demonstrated a proof-of-concept, the next step would certainly be making use of HPC in order to speed up research progress. The problem of computational intensity can also be viewed through the lens of implementation: it might be worthwhile to test MDP models that do not scale with the number of vehicles, as seen in [55]. Going a step further, Python is not the best programming language in terms of speed, although its accessibility must not be neglected. Major data handling operations within the code of FleetRL were already optimized for computational speed, but further improvements could be made by wrapping the code in C or C++, for example via Cython [114]. Finally, modelling a realistic EV charging problem as an MDP was a challenging task. A learning from this project was therefore to always weigh RL against other approaches that might be sufficient to solve the problem at hand. In this case, given the results and that the boundaries of linear optimization were reached within this study, it was considered an effort worth undertaking.

Diving into the learning process of the RL agent, it was found that not all intricacies of the problem were perfectly recognized and solved. On one hand, this was due to the difficulty of the problem: the reward function was sparse, meaning that the reward signals were rare but important. Multiple decisions lead up to the moment of vehicle departure, where a penalty was either given or not. Additionally, intraday price differences had to be recognized that were hours apart. Properly adjusting the policy according to these signals was therefore inherently difficult, and partially explains the high computational intensity. However, there are a few improvements to be made. One potential improvement is the introduction of expert demonstrations, or imitation learning, which was applied in the EV charging context by [115]. Previously compiled ”good” decisions were shown to the agent, acting as a guidance during the exploration phase. This potentially avoids the scenario that a beneficial strategy is never discovered due to chance. Another potential improvement could be testing different model-free RL agents on the environment. In this study, SAC, TD3, DDPG, and PPO were tested;

only PPO was found to develop an optimal strategy. New models continue to emerge, and their architectures might be better suited for the problem, potentially bringing improvements.

It also needs to be said that the use of model-free RL methods might pose a potential limitation to certain applications. This is because model-free methods remain entirely black-box - and are therefore not explainable. Model-free RL methods come with flexibility and are able to discover new optima (c.f. DeepMind and AlphaGo [50]), but can also introduce an unacceptable degree of uncertainty. Taking the example of transmission system operators, simple, explainable models are often preferred for forecasting, even though they could easily be outperformed by deep learning models. This is because some applications are simply too sensitive to be steered by a black-box algorithm. Potential remedies exist: [116] introduces a safety layer that is wrapped around the actions of the model-free agent, ensuring that vehicles are always fully charged upon departure. Finally, the transition from model-free to model-based RL-methods could yield a degree of explainability that might be required in some contexts. Promising methods thereby include approaches that learn the problem dynamics through experience [117–120], or receive the physics representation prior to training [121, 122].

Finally, several implementation aspects were identified as potential limitations and could be improved upon in future work. The LP model was relatively simple, and did not maximize the potential of linear optimization. Although in this study it did not yield practical results to optimize a reward function instead of fulfilling constraints, this approach can be viable and should be further investigated. Further, Model-Predictive Control (MPC) could make a linear optimization more realistic, as a rolling forecast window is used to compute optimizations instead of a year of perfect knowledge. This however introduces new challenges such as computational intensity and forecasting accuracy. In the end, building the simple LP model provided useful insights into applying linear optimization in tariff scenarios: the caretaker use-case could not be solved feasibly, and the model had to be reworked before a solution could be found. In realistic and dynamic environments, LP models can therefore fall short because of their perfect knowledge requirement and the inability to deal with uncertainty or changing circumstances.

As for the business case analysis, scaling up the results of the 1 EV agent to match the

number of vehicles for the three use-cases poses a potential limitation. It is not certain that the same economic performance will be reached when dozens of cars are optimized at the same time. This has therefore been identified as a point for future work. Additionally, vehicle-to-grid, and other flexibility options could be integrated into FleetRL, making it more applicable to a wider range of use-cases. Adding additional appliances such as heat pumps and stationary battery storage systems would further increase the agent's potential to leverage flexibility and increase savings.

Overall, the results of this study confirm the literature claims that RL is a viable solution in the domain of EV charging. This confirmation is given after testing RL on three commercial use-cases, taking into account investments and third parties that are required to make smart charging possible. FleetRL, the RL environment developed in this study, thereby includes non-linear battery degradation, schedule generation, spot market prices, PV and building load data, as well as numerous features to avoid simplifications found in other openly-available EV charging environments [91–94]. To date, FleetRL is thus the most elaborate open-source RL environment framework available. During development, a modular approach was taken, allowing for easy customizability, addition of new use-cases, and improvement of the framework. Going forward, FleetRL can hopefully be used by RL-researchers to benchmark RL agents for EV charging, making it possible to not only assess a reward signal, but the economic feasibility in the real world.

Chapter 7

Conclusion

This study applied Deep Reinforcement Learning to optimize EV charging for commercial vehicle fleets and reported encouraging findings regarding real-world applicability and economic feasibility. Two main research questions motivated the study and its efforts:

1. *What is the potential of RL-based EV charging in terms of cost savings and real-world applicability when compared to the most common EV charging strategies (e.g. uncontrolled charging and optimization-based smart charging)?*
2. *How do the three investigated commercial use-cases differ regarding their cost savings and potential to leverage RL-based charging optimization?*

The research questions were tackled with a methodological approach to design commercial use-cases and a corresponding Markov Decision Process model to solve the problem with RL. The software framework that was developed, FleetRL, will be published open-source together with this study. It currently is the most realistic open-source RL environment for commercial vehicle fleets, with features such as non-linear battery degradation and probabilistic vehicle schedule generation. The framework was used to obtain the results of this study.

Compared to commonly used charging strategies, such as uncontrolled charging, night charging, or distributed charging, RL performed remarkably well: savings of up to 85% were achieved, and grid connection overloadings were almost entirely avoided. A linear optimization with one year of perfect knowledge served as an additional benchmark to indicate what a best-case scenario could look like. While this arguably unfair

comparison outperformed RL in most disciplines, it revealed practical limitations of LP in realistic problems with uncertainty and tight constraints that did not occur with RL - a method that only used real-time information. With a time resolution of 15 minutes and without perfect knowledge, RL effectively avoided grid overloadings and reduced charging costs. Charging costs were thereby reduced by such an extent that it was economically feasible to upgrade charging infrastructure and pay third party service providers for the load management service. The economic analysis further revealed that all three commercial use-cases achieved positive net-present values within 10 years. The first research question can therefore be answered with a strong affirmation: When compared to the most common charging strategies, RL showed savings of up to 85%, and proved its real-time applicability in a complex, real-time environment.

Comparing the different use-cases, it was observed that use-cases with larger battery sizes yielded larger savings. However, when investigating the energy-specific savings in €$/kWh$, it was found that all three use-cases achieved essentially the same specific savings. This indicates that the load profiles, schedules, or vehicle types did not lead to an inherent difficulty to generate savings for one specific use-case. Therefore, the second research question can be answered with a promising affirmation that there is a potential for RL-based charging optimization for last-mile delivery, utility, and caretaker use-cases alike. The considerably different electrification rates likely originate from different technology adoption rates, regulatory incentives and technological expertise: A utility company is at the forefront of developments of the electricity network and therefore likely to be an early adopter. Similarly, a last-mile delivery company has vehicles at the heart of its business model and is therefore more sensitive to technological changes than a caretaker company. The answer to the second research question is promising, because it implies that all three company types can potentially engage in load balancing activities with their fleets and generate an economically beneficial result compared to the status quo.

Besides the promising results, practical limitations of RL were discovered. First, the training process was computationally expensive. It poses a potential bottleneck for anyone looking to apply RL in complex environments and needs to be taken into account in the resource planning stage. Second, model-free RL might introduce unacceptable uncertainties to critical domains where explainability is required. Model-based RL with an embedded physics representation is a potential solution to this issue.

Further, the methodological approach with its scope and assumptions leaves room for future additions. Non-scaling MDP representations could be investigated to increase computational efficiency for multiple vehicles. Additionally, vehicle-to grid was only considered in the form of energy arbitrage. Balancing services could be implemented to extend the possible revenue streams of the fleet. Adding additional flexible appliances such as heat pumps or stationary battery storage systems could further increase the flexibility and economic potential. Furthermore, load management solutions do not only aim to save money, but also try to integrate as many EVs and renewable energy sources into the grid as possible. This environmental impact could be investigated by analysing the effect of the RL charging strategy on the distribution grid.

Given the milestones achieved in recent years, it is likely that Reinforcement Learning will play an increasingly important role in the energy transition, as well as human interactions with intelligent systems in general. It is therefore of paramount importance that an understanding and a consensus are built regarding where these systems can be safely deployed - and where they cannot. Wherever real-world experiments are too costly or dangerous, realistic simulations gain importance. This thesis and its emerging framework FleetRL aim to contribute to this research domain by providing a platform to perform realistic studies on RL applications in EV charging.

Bibliography

[1] IEA, *CO2 emissions – Global Energy Review 2021 – Analysis*, en-GB, 2021. [Online]. Available: https://www.iea.org/reports/global-energy-review-2021/co2-emissions (visited on 01/12/2023).

[2] IEA, *Global energy-related CO2 emissions by sector – Charts – Data & Statistics*, en-GB. [Online]. Available: https://www.iea.org/data-and-statistics/charts/global-energy-related-co2-emissions-by-sector (visited on 01/12/2023).

[3] IEA, *Transport – Topics*, en-GB, 2022. [Online]. Available: https://www.iea.org/topics/transport (visited on 01/12/2023).

[4] European Commission, *Transport electrification (ELT) | TRIMIS*. [Online]. Available: http://trimis.ec.europa.eu/roadmaps/transport-electrification-elt (visited on 01/16/2023).

[5] Akaber, P., Hughes, T., and Sobolev, S., "MILP-based customer-oriented E-Fleet charging scheduling platform," *IET Smart Grid*, vol. 4, no. 2, pp. 215–223, 2021, ISSN: 2515-2947. DOI: 10.1049/stg2.12034. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1049/stg2.12034 (visited on 01/16/2023).

[6] Faddel, S., Al-Awami, A. T., and Mohammed, O. A., "Charge Control and Operation of Electric Vehicles in Power Grids: A Review," *Energies*, vol. 11, no. 4, p. 701, Apr. 2018, Number: 4 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 1996-1073. DOI: 10.3390/en11040701. [Online]. Available: https://www.mdpi.com/1996-1073/11/4/701 (visited on 01/12/2023).

[7] Franco, J. F., Rider, M. J., and Romero, R., "A Mixed-Integer Linear Programming Model for the Electric Vehicle Charging Coordination Problem in Unbalanced Electrical Distribution Systems," *IEEE Transactions on Smart Grid*, vol. 6, no. 5, pp. 2200–2210, Sep. 2015, Conference Name: IEEE Transactions on Smart Grid, ISSN: 1949-3061. DOI: 10.1109/TSG.2015.2394489.

[8] Abdullah, H. M., Gastli, A., and Ben-Brahim, L., "Reinforcement Learning Based EV Charging Management Systems–A Review," *IEEE Access*, vol. 9, pp. 41 506–41 531, 2021, Conference Name: IEEE Access, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3064354.

[9] Cording, E., *EnzoCording/FleetRL*, original-date: 2023-03-23T09:33:53Z, Aug. 2023. [Online]. Available: https://github.com/EnzoCording/FleetRL (visited on 08/14/2023).

[10] Hecht, C., Spreuer, K. G., Figgener, J., and Sauer, D. U., "Market Review and Technical Properties of Electric Vehicles in Germany," *Vehicles*, vol. 4, no. 4, pp. 903–916, Dec. 2022, Number: 4 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2624-8921. DOI: 10.3390/vehicles4040049. [Online]. Available: https://www.mdpi.com/2624-8921/4/4/49 (visited on 08/07/2023).

[11] Federal Motor Transport Authority and NOW GmbH, "KBA-Report April 2022, 2023," Official vehicle statistics, May 2023. [Online]. Available: https://www.now-gmbh.de/wissensfinder/ (visited on 06/08/2023).

[12] ENTSO-E, *ENTSO-E Transparency Platform*, 2023. [Online]. Available: https://transparency.entsoe.eu/ (visited on 07/05/2023).

[13] Bundesnetzagentur, *Ladesäulenkarte*, Aug. 2023. [Online]. Available: https://www.bundesnetzagentur.de/DE/Fachthemen/ElektrizitaetundGas/E-Mobilitaet/Ladesaeulenkarte/Karte/Ladesaeulenkarte.html?nn=266476 (visited on 08/07/2023).

[14] Hecht, C., Figgener, J., and Sauer, D. U., "Analysis of electric vehicle charging station usage and profitability in Germany based on empirical data," *iScience*, vol. 25, no. 12, p. 105 634, Dec. 2022, ISSN: 25890042. DOI: 10.1016/j.isci.2022.105634. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S258900422201906X (visited on 08/07/2023).

[15] BMWK-Federal Ministry for Economics Affairs and Climate, *Elektromobilität in Deutschland*, Dec. 2022. [Online]. Available: https://www.bmwk.de/Redaktion/DE/Dossier/elektromobilitaet.html (visited on 08/07/2023).

[16] Federal Ministry of Justice, *LSV - Verordnung über technische Mindestanforderungen an den sicheren und interoperablen Aufbau und Betrieb von öffentlich zugänglichen Ladepunkten für elektrisch betriebene Fahrzeuge 1*, German, Mar. 2016. [Online]. Available: https://www.gesetze-im-internet.de/lsv/BJNR045700016.html (visited on 08/07/2023).

[17] ADAC, *Förderung für Elektroautos*, Jul. 2023. [Online]. Available: https://www.adac.de/rund-ums-fahrzeug/elektromobilitaet/kaufen/foerderung-elektroautos/ (visited on 08/07/2023).

[18] Federal Ministry of Justice, *EStG - Einkommenssteuergesetz*, Oct. 2009. [Online]. Available: https://www.gesetze-im-internet.de/estg/ (visited on 08/07/2023).

[19] KfW - Kreditanstalt für Wiederaufbau, *Merkblatt: Ladestationen für Elektrofahrzeuge – Unternehmen*, Jul. 2022. [Online]. Available: https://www.kfw.de/PDF/Download-Center/F%C3%B6rderprogramme-(Inlandsf%C3%B6rderung)/PDF-Dokumente/6000004939_M_441.pdf (visited on 08/14/2023).

[20] Ministry of Economic Northrhine Westfalia, *Förderung für Unternehmen | Elektromobilität.NRW*, 2023. [Online]. Available: https://www.elektromobilitaet.nrw/unternehmen/foerderung-fuer-unternehmen/ (visited on 08/07/2023).

[21] Federal Government of Germany, *Intelligente Strommessung für die Energiewende | Bundesregierung*, Jun. 2023. [Online]. Available: https://www.bundesregierung.de/breg-de/schwerpunkte/klimaschutz/digitale-energiewende-2157184 (visited on 08/07/2023).

[22] Tibber, *Strompreise*, 2023. [Online]. Available: https://tibber.com/de/strompreise (visited on 07/06/2023).

[23] aWATTar, *aWATTar HOURLY*, 2023. [Online]. Available: https://www.awattar.de/tariffs/hourly (visited on 07/06/2023).

[24] The Mobility House, *ChargePilot Lade- und Energiemanagement*, 2023. [Online]. Available: https://www.mobilityhouse.com/de_de/lade-und-energiemanagement-paket-core.html (visited on 08/07/2023).

[25] Ridergy, *RiDERgy: Efficient Fleet Electrification & Charging*, 2023. [Online]. Available: https://ridergy.com/ (visited on 08/07/2023).

[26] E.ON, *Lastmanagement für Ihr Unternehmen | E.ON Drive*, 2023. [Online]. Available: https://www.eon-drive.de/de/loesungen/lastmanagement.html (visited on 08/07/2023).

[27] ABB, *Load Management Devices*, 2023. [Online]. Available: https://new.abb.com/low-voltage/products/system-pro-m/control-and-automation-devices/load-management-devices (visited on 08/07/2023).

[28] Alliance, O. C., *OCPP*, 2023. [Online]. Available: https://www.openchargealliance.org/protocols/ocpp-201/ (visited on 08/07/2023).

[29] ISO, *ISO 15118-20:2022*, Apr. 2022. [Online]. Available: https://www.iso.org/standard/77845.html (visited on 08/07/2023).

[30] EEBus, *EEBus*, en-US, 2023. [Online]. Available: https://www.eebus.org/ (visited on 08/07/2023).

[31] MQTT, *MQTT - The Standard for IoT Messaging*, 2023. [Online]. Available: https://mqtt.org/ (visited on 08/07/2023).

[32] Revolution Pi, *Industrie PC auf Raspberry Pi Basis überblick | Industrial Raspberry Pi - Revolution Pi*, Jan. 2018. [Online]. Available: https://revolutionpi.de/revolution-pi-serie (visited on 08/07/2023).

[33] FLEXeCharge, *Smart Charging Gateway*, 2022. [Online]. Available: https://flexecharge.com/wp-content/uploads/2022/06/Smart-Charge-Gateway_Product-Brochure.pdf (visited on 08/14/2023).

[34] Yilmaz, M. and Krein, P. T., "Review of the Impact of Vehicle-to-Grid Technologies on Distribution Systems and Utility Interfaces," *IEEE Transactions on Power Electronics*, vol. 28, no. 12, pp. 5673–5689, Dec. 2013, ISSN: 0885-8993, 1941-0107. DOI: 10.1109/TPEL.2012.2227500. [Online]. Available: https://ieeexplore.ieee.org/document/6353961/ (visited on 08/02/2023).

[35] Masoum, M. A., Moses, P. S., and Hajforoosh, S., "Distribution transformer stress in smart grid with coordinated charging of Plug-In Electric Vehicles," in *2012 IEEE PES Innovative Smart Grid Technologies (ISGT)*, Jan. 2012, pp. 1–8. DOI: 10.1109/ISGT.2012.6175685.

[36] US Department of Energy, *EV Charging Tariff, Minnesota Statutes 216B.1614*, May 2014. [Online]. Available: https://afdc.energy.gov/laws/11483 (visited on 08/05/2023).

[37] Deilami, S., Masoum, A. S., Moses, P. S., and Masoum, M. A. S., "Real-Time Coordination of Plug-In Electric Vehicle Charging in Smart Grids to Minimize Power Losses and Improve Voltage Profile," *IEEE Transactions on Smart Grid*, vol. 2, no. 3, pp. 456–467, Sep. 2011, ISSN: 1949-3053, 1949-3061. DOI: 10.1109/TSG.2011.2159816. [Online]. Available: http://ieeexplore.ieee.org/document/5986769/ (visited on 08/02/2023).

[38] Qiu, D., Wang, Y., Hua, W., and Strbac, G., "Reinforcement learning for electric vehicle applications in power systems:A critical review," *Renewable and Sustainable Energy Reviews*, vol. 173, p. 113 052, Mar. 2023, ISSN: 1364-0321. DOI: 10.1016/j.rser.2022.113052. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032122009339 (visited on 01/11/2023).

[39] Arias, N. B., Franco, J. F., Lavorato, M., and Romero, R., "Metaheuristic optimization algorithms for the optimal coordination of plug-in electric vehicle charging in distribution systems with distributed generation," *Electric Power Systems Research*, vol. 142, pp. 351–361, Jan. 2017, ISSN: 0378-7796. DOI: 10.1016/j.epsr.2016.09.018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378779616303704 (visited on 08/05/2023).

[40] Hutterer, S., Auinger, F., and Affenzeller, M., "Metaheuristic Optimization of Electric Vehicle Charging Strategies in Uncertain Environment," Jun. 2012.

[41] Yang, J., He, L., and Fu, S., "An improved PSO-based charging strategy of electric vehicles in electrical distribution grid," *Applied Energy*, vol. 128, pp. 82–92, Sep. 2014, ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2014.04.047. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261914004000 (visited on 08/05/2023).

[42] Nerkar, M., Mukherjee, A., and Soni, B. P., "A review on optimization scheduling methods of charging and discharging of EV," *AIP Conference Proceedings*, vol. 2452, no. 1, p. 040 002, Nov. 2022, Publisher: American Institute of Physics, ISSN: 0094-243X. DOI: 10.1063/5.0114625. [Online]. Available: https://aip.scitation.org/doi/abs/10.1063/5.0114625 (visited on 01/12/2023).

[43] Tesauro, G., "Temporal difference learning and TD-Gammon," *Communications of the ACM*, vol. 38, no. 3, pp. 58–68, Mar. 1995, ISSN: 0001-0782. DOI: 10.1145/203330.203343. [Online]. Available: https://doi.org/10.1145/203330.203343 (visited on 03/07/2023).

[44] Baird, L., "Residual Algorithms: Reinforcement Learning with Function Approximation," in *Machine Learning Proceedings 1995*, A. Prieditis and S. Russell, Eds., San Francisco (CA): Morgan Kaufmann, Jan. 1995, pp. 30–37, ISBN: 978-1-55860-377-6. DOI: 10.1016/B978-1-55860-377-6.50013-X. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B978155860377650013X (visited on 03/07/2023).

[45] Boyan, J. and Moore, A., "Generalization in Reinforcement Learning: Safely Approximating
the Value Function," in *Advances in Neural Information Processing Systems*, vol. 7, MIT Press, 1994. [Online]. Available: https://proceedings.neurips.cc/paper/1994/hash/ef50c335cca9f340bde656363ebd02fd-Abstract.html (visited on 03/07/2023).

[46] Sutton, R. S. and Barto, A. G., *Reinforcement learning: an introduction* (Adaptive computation and machine learning series), Second edition. Cambridge, Massachusetts: The MIT Press, 2018, ISBN: 978-0-262-03924-6.

[47] Stanford University, *CS234: Reinforcement Learning Winter 2023*. [Online]. Available: https://web.stanford.edu/class/cs234/modules.html (visited on 03/16/2023).

[48] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M., *Playing Atari with Deep Reinforcement Learning*, arXiv:1312.5602 [cs], Dec. 2013. [Online]. Available: http://arxiv.org/abs/1312.5602 (visited on 03/07/2023).

[49] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015, Number: 7540 Publisher: Nature Publishing Group, ISSN: 1476-4687. DOI: 10.1038/nature14236. [Online]. Available: https://www.nature.com/articles/nature14236 (visited on 02/11/2023).

[50] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Driessche, G. van den, Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D., "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016, Number: 7587 Publisher: Nature Publishing Group, ISSN: 1476-4687. DOI: 10.1038/nature16961. [Online]. Available: https://www.nature.com/articles/nature16961 (visited on 08/10/2023).

[51] Gu, S., Yang, L., Du, Y., Chen, G., Walter, F., Wang, J., Yang, Y., and Knoll, A., *A Review of Safe Reinforcement Learning: Methods, Theory and Applications*, arXiv:2205.10330 [cs], Feb. 2023. DOI: 10.48550/arXiv.2205.10330. [Online]. Available: http://arxiv.org/abs/2205.10330 (visited on 03/20/2023).

[52] Web of Science, *Number of Papers published on RL in EV Charging*. [Online]. Available: https://www.webofscience.com/wos/woscc/analyze-results/3402e6ef-dcf4-49f3-bb79-767b4ce4e8f5-78b33808 (visited on 03/14/2023).

[53] Valogianni, K., Ketter, W., and Collins, J., "Smart charging of electric vehicles using reinforcement learning," *AAAI Workshop - Technical Report*, pp. 41–48, Jan. 2013.

[54] Valogianni, K., Ketter, W., Collins, J., and Zhdanov, D., "Effective Management of Electric Vehicle Storage Using Smart Charging," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 28, no. 1, Jun. 2014, ISSN: 2374-3468, 2159-5399. DOI: 10.1609/aaai.v28i1.8760. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/8760 (visited on 03/14/2023).

[55] Sadeghianpourhamami, N., Deleu, J., and Develder, C., "Definition and Evaluation of Model-Free Coordination of Electrical Vehicle Charging With Reinforcement Learning," *IEEE Transactions on Smart Grid*, vol. 11, no. 1, pp. 203–214, Jan. 2020, Conference Name: IEEE Transactions on Smart Grid, ISSN: 1949-3061. DOI: 10.1109/TSG.2019.2920320.

[56] Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., and Dormann, N., "Stable-Baselines3: Reliable Reinforcement Learning Implementations," 2021.

[57] Markov, A. A., "Extension of the law of large numbers to dependent events," Ru, *Bulletin of the Society of the Physics Mathematics*, vol. 2, pp. 155–156, 1906.

[58] Bellman, R. and Dreyfus, S., *Dynamic programming* (Princeton Landmarks in mathematics), eng, 1. Princeton Landmarks in Mathematics ed., with a new introduction. Princeton, NJ: Princeton University Press, 2010, ISBN: 978-0-691-14668-3.

[59] Stanford University, *REINFORCEjs: Gridworld with Dynamic Programming*. [Online]. Available: https://cs.stanford.edu/people/karpathy/reinforcejs/gridworld_dp.html (visited on 03/17/2023).

[60] Watkins, C. J. C. H. and Dayan, P., "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, May 1992, ISSN: 0885-6125, 1573-0565. DOI: 10.1007/BF00992698. [Online]. Available: http://link.springer.com/10.1007/BF00992698 (visited on 07/11/2023).

[61] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O., "Proximal Policy Optimization Algorithms," 2017, Publisher: arXiv Version Number: 2. DOI: 10.48550/ARXIV.1707.06347. [Online]. Available: https://arxiv.org/abs/1707.06347 (visited on 06/26/2023).

[62] Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S., "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," 2018, Publisher: arXiv Version Number: 2. DOI: 10.48550/ARXIV.1801.01290. [Online]. Available: https://arxiv.org/abs/1801.01290 (visited on 07/13/2023).

[63] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D., "Continuous control with deep reinforcement learning," 2015, Publisher: arXiv Version Number: 6. DOI: 10.48550/ARXIV.1509.02971. [Online]. Available: https://arxiv.org/abs/1509.02971 (visited on 07/13/2023).

[64] Goodfellow, I., Bengio, Y., and Courville, A., *Deep learning* (Adaptive computation and machine learning). Cambridge, Massachusetts: The MIT Press, 2016, ISBN: 978-0-262-03561-3.

[65] Rumelhart, D. E., Hinton, G. E., and Williams, R. J., "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986, ISSN: 0028-0836, 1476-4687. DOI: 10.1038/323533a0. [Online]. Available: https://www.nature.com/articles/323533a0 (visited on 07/13/2023).

[66] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P., "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, ISSN: 00189219. DOI: 10.1109/5.726791. [Online]. Available: http://ieeexplore.ieee.org/document/726791/ (visited on 07/13/2023).

[67] Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M., "Deterministic policy gradient algorithms," in *Proceedings of the 31st International Conference on Machine Learning*, E. P. Xing and T. Jebara, Eds., ser. Proceedings of Machine Learning Research, vol. 32, Bejing, China: PMLR, 22–24 Jun 2014, pp. 387–395. [Online]. Available: https://proceedings.mlr.press/v32/silver14.html.

[68] Fujimoto, S., Hoof, H. van, and Meger, D., "Addressing Function Approximation Error in Actor-Critic Methods," 2018, Publisher: arXiv Version Number: 3. DOI: 10.48550/ARXIV.1802.09477. [Online]. Available: https://arxiv.org/abs/1802.09477 (visited on 06/26/2023).

[69] Huang, S., Dossa, R. F. J., Raffin, A., Kanervisto, A., and Wang, W., *The 37 Implementation Details of Proximal Policy Optimization*, Mar. 2022. [Online]. Available: https://iclr-blog-track.github.io/2022/03/25/ppo-implementation-details/.

[70] Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P., *High-Dimensional Continuous Control Using Generalized Advantage Estimation*, arXiv:1506.02438 [cs], Oct. 2018. DOI: 10.48550/arXiv.1506.02438. [Online]. Available: http://arxiv.org/abs/1506.02438 (visited on 07/26/2023).

[71] Gaete-Morales, C., Kramer, H., Schill, W.-P., and Zerrahn, A., "An open tool for creating battery-electric vehicle time series from empirical data, emobpy," *Scientific Data*, vol. 8, no. 1, p. 152, Jun. 2021, Number: 1 Publisher: Nature Publishing Group, ISSN: 2052-4463. DOI: 10.1038/s41597-021-00932-9. [Online]. Available: https://www.nature.com/articles/s41597-021-00932-9 (visited on 06/06/2023).

[72] Gaete-Morales, C., *Emobpy: Application for the German case*, eng, Feb. 2021. DOI: 10.5281/zenodo.4514928. [Online]. Available: https://zenodo.org/record/4514928 (visited on 01/10/2023).

[73] Electrive, *Electrive.net*. [Online]. Available: https://www.electrive.net/ (visited on 06/06/2023).

[74] Wilson, E., Parker, A., Fontanini, A., Present, E., Reyna, J., Adhikari, R., Bianchi, C., CaraDonna, C., Dahlhausen, M., Kim, J., LeBar, A., Liu, L., Praprost, M., Zhang, L., DeWitt, P., Merket, N., Speake, A., Hong, T., Li, H., Mims Frick, N., Wang, Z., Blair, A., Horsey, H., Roberts, D., Trenbath, K., Adekanye, O., Bonnema, E., El Kontar, R., Gonzalez, J., Horowitz, S., Jones, D., Muehleisen, R., Platthotam, S., Reynolds, M., Robertson, J., Sayers, K., and Li, Q., *End-Use Load Profiles for the U.S. Building Stock*, 2021. DOI: 10.25984/1876417. [Online]. Available: https://www.osti.gov/servlets/purl/1876417/ (visited on 06/06/2023).

[75] Department of Energy, *Commercial Reference Buildings*. [Online]. Available: https://www.energy.gov/eere/buildings/commercial-reference-buildings (visited on 05/30/2023).

[76] World Data, *Climate comparison: Germany / Washington*. [Online]. Available: https://www.worlddata.info/climate-comparison.php?r1=germany&r2=us-washington (visited on 06/06/2023).

[77] Pfenninger, S. and Staffell, I., "Long-term patterns of European PV output using 30 years of validated hourly reanalysis and satellite data," *Energy*, vol. 114, pp. 1251–1265, Nov. 2016, ISSN: 03605442. DOI: 10.1016/j.energy.2016.08.060. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0360544216311744 (visited on 06/06/2023).

[78] Fairsenden, *Fairsenden – Your time Fair delivery*, en-US, Jun. 2023. [Online]. Available: https://fairsenden.digital/en/ (visited on 06/06/2023).

[79] Johansson, E. and Rostmark, M., *Electrification of Last-Mile Transport : A Study of Charging Infrastructure and Collaborative Business Model*, eng. 2022. [Online]. Available: https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-310914 (visited on 05/30/2023).

[80] Mercedes Benz, *eVito Kastenwagen*. [Online]. Available: https://www.mercedes-benz.de/vans/models/evito/panel-van/overview.html (visited on 06/12/2023).

[81] LC, *Our projects*, en-US. [Online]. Available: https://www.lc.se/en/our-projects/ (visited on 06/07/2023).

[82] Urbanized, *Overcoming the roadblocks: Electric vehicle adoption in small and large logistics companies - URBANIZED - modUlaR and flexible solutions for urBAN-sIzed Zero-Emissions last-mile Delivery and services vehicles*, en-US, May 2023. [Online]. Available: https://urbanized.eu/overcoming-the-roadblocks-electric-vehicle-adoption-in-small-and-large-logistics-companies/ (visited on 06/05/2023).

[83] Amazon, *Our Carbon Footprint*. [Online]. Available: https://sustainability.aboutamazon.com/environment/carbon-footprint (visited on 06/12/2023).

[84] DHL, *Zero emissions by 2050: DHL announces ambitious new environmental protection target*, en-Go. [Online]. Available: https://www.dhl.com/global-en/delivered/sustainability/zero-emissions-by-2050.html (visited on 06/12/2023).

[85] Schwierz, P., *Mark Oswald über Elektromobilität in Pflegediensten - electrive.net*, Jun. 2018. [Online]. Available: https://www.electrive.net/2018/06/28/mark-oswald-ueber-elektromobilitaet-in-pflegediensten (visited on 06/07/2023).

[86] Schaal, S., *Pflegedienst MobiDoc beschafft 30 Smart EQ - electrive.net*, Sep. 2022. [Online]. Available: https://www.electrive.net/2022/08/26/pflegedienst-mobidoc-beschafft-30-smart-eq (visited on 06/07/2023).

[87] Mercedes Benz, *Smart EQ fortwo | Der vollelektrische Zweisitzer*. [Online]. Available: https://www.smart.mercedes-benz.com/de/de/modelle/smart-eq-fortwo-coupe (visited on 06/12/2023).

[88] Vattenfall, *Vattenfall is switching its whole car fleet to electric vehicles*, Feb. 2017. [Online]. Available: https://group.vattenfall.com/press-and-media/pressreleases/2017/vattenfall-is-switching-its-whole-car-fleet-to-electric-vehicles (visited on 07/20/2023).

[89] Hausmann, F., *Vattenfall: Energiekonzern stellt auf E-Mobilität um*, Section: StatischeSeiten, Apr. 2017. [Online]. Available: https://www.firmenauto.de/vattenfall-vattenfall-setzt-ganz-auf-e-mobilitaet-8897131.html (visited on 06/07/2023).

[90] Citroen, *Citroën Berlingo Kastenwagen | Flexibler Profi-Transporter*. [Online]. Available: https://business.citroen.de/modellpalette/berlingo-kastenwagen.html (visited on 06/12/2023).

[91] Lee, Z. J., "The Adaptive Charging Network Research Portal: Systems, Tools, and Algorithms," Medium: PDF Version Number: Final, Ph.D. dissertation, California Institute of Technology, Jun. 2021. DOI: 10.7907/8EQG-E110. [Online]. Available: https://resolver.caltech.edu/CaltechTHESIS:05282021-174411678 (visited on 06/08/2023).

[92] Karatzinis, G., Korkas, C., Terzopoulos, M., Tsaknakis, C., Stefanopoulou, A., Michailidis, I., and Kosmatopoulos, E., "Chargym: An ev charging station model for controller benchmarking," in *IFIP International Conference on Artificial Intelligence Applications and Innovations*, Springer, 2022, pp. 241–252.

[93] Gallego, F., Martín, C., Díaz, M., and Garrido, D., "Maintaining flexibility in smart grid consumption through deep learning and deep reinforcement learning," *Energy and AI*, vol. 13, p. 100 241, Jul. 2023, ISSN: 2666-5468. DOI: 10.1016/j.egyai.2023.100241. [Online]. Available: https://www.

sciencedirect.com/science/article/pii/S2666546823000137 (visited on 06/12/2023).

[94] Sharma, S., *Gym-acnportal*, original-date: 2020-04-03T16:34:13Z, Sep. 2022. [Online]. Available: https://github.com/caltech-netlab/gym-acnportal (visited on 06/12/2023).

[95] ebalanceplus, *Ebalanceplus – Smart energy flexibility for distribution grids*. [Online]. Available: https://www.ebalanceplus.eu/ (visited on 06/12/2023).

[96] Xu, B., Oudalov, A., Ulbig, A., Andersson, G., and Kirschen, D. S., "Modeling of Lithium-Ion Battery Degradation for Cell Life Assessment," *IEEE Transactions on Smart Grid*, vol. 9, no. 2, pp. 1131–1140, Mar. 2018, Conference Name: IEEE Transactions on Smart Grid, ISSN: 1949-3061. DOI: 10.1109/TSG.2016.2578950.

[97] BDEW, *BDEW-Strompreisanalyse April 2023*, Apr. 1, 2023. [Online]. Available: https://www.bdew.de/service/daten-und-grafiken/bdew-strompreisanalyse/ (visited on 07/10/2023).

[98] Federal Ministry of Justice, *EEG §48 Abs. 6*, Jul. 2014. [Online]. Available: https://www.gesetze-im-internet.de/eeg_2014/BJNR106610014.html (visited on 07/06/2023).

[99] Chem, L., *Rechargeable Lithium Ion Battery E63*, en-gb, Feb. 2018. [Online]. Available: http://queenbattery.com.cn/our-products/677-lg-e63-376v-63ah-li-po-li-polymer-battery-cell.html?search_query=lg+e63&results=1 (visited on 06/13/2023).

[100] Vetter, J., Novák, P., Wagner, M. R., Veit, C., Möller, K. .-., Besenhard, J. O., Winter, M., Wohlfahrt-Mehrens, M., Vogler, C., and Hammouche, A., "Ageing mechanisms in lithium-ion batteries," *Journal of Power Sources*, vol. 147, no. 1, pp. 269–281, Sep. 2005, ISSN: 0378-7753. DOI: 10.1016/j.jpowsour.2005.01.006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378775305000832 (visited on 06/13/2023).

[101] Han, X., Ouyang, M., Lu, L., Li, J., Zheng, Y., and Li, Z., "A comparative study of commercial lithium ion battery cycle life in electrical vehicle: Aging mechanism identification," *Journal of Power Sources*, vol. 251, pp. 38–54, Apr. 2014, ISSN: 0378-7753. DOI: 10.1016/j.jpowsour.2013.11.029.

[Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378775313018569 (visited on 06/13/2023).

[102] Laresgoiti, I., Käbitz, S., Ecker, M., and Sauer, D. U., "Modeling mechanical degradation in lithium ion batteries during cycling: Solid electrolyte interphase fracture," *Journal of Power Sources*, vol. 300, pp. 112–122, Dec. 2015, ISSN: 0378-7753. DOI: 10.1016/j.jpowsour.2015.09.033. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378775315302949 (visited on 06/13/2023).

[103] Santner, H. J., Möller, K. .-., Ivančo, J., Ramsey, M. G., Netzer, F. P., Yamaguchi, S., Besenhard, J. O., and Winter, M., "Acrylic acid nitrile, a film-forming electrolyte component for lithium-ion batteries, which belongs to the family of additives containing vinyl groups," *Journal of Power Sources*, Selected papers presented at the 11th International Meeting on Lithium Batteries, vol. 119-121, pp. 368–372, Jun. 2003, ISSN: 0378-7753. DOI: 10.1016/S0378-7753(03)00268-4. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378775303002684 (visited on 06/13/2023).

[104] Smart, M. C., Ratnakumar, B. V., Ryan-Mowrey, V. S., Surampudi, S., Prakash, G. K. S., Hu, J., and Cheung, I., "Improved performance of lithium-ion cells with the use of fluorinated carbonate-based electrolytes," *Journal of Power Sources*, Selected papers presented at the 11th International Meeting on Lithium Batteries, vol. 119-121, pp. 359–367, Jun. 2003, ISSN: 0378-7753. DOI: 10.1016/S0378-7753(03)00266-0. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378775303002660 (visited on 06/13/2023).

[105] Janiszewski, P., *Rainflow: Implementation of ASTM E1049-85 rainflow cycle counting algorithm*. [Online]. Available: https://github.com/iamlikeme/rainflow (visited on 06/13/2023).

[106] ASTM, *Standard Practices for Cycle Counting in Fatigue Analysis*. [Online]. Available: https://www.astm.org/e1049-85r17.html (visited on 06/13/2023).

[107] EPEX Spot, *Single Day-Ahead Coupling*, Jun. 2023. [Online]. Available: https://www.epexspot.com/sites/default/files/download_center_

files/Day-Ahead%20MRC%20Processes%20%2802.07.2019%29.pdf (visited on 07/20/2023).

[108] Bengio, Y., Louradour, J., Collobert, R., and Weston, J., "Curriculum learning," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09, New York, NY, USA: Association for Computing Machinery, Jun. 2009, pp. 41–48, ISBN: 978-1-60558-516-1. DOI: 10.1145/1553374.1553380. [Online]. Available: https://doi.org/10.1145/1553374.1553380 (visited on 06/14/2023).

[109] Sukhbaatar, S., Lin, Z., Kostrikov, I., Synnaeve, G., Szlam, A., and Fergus, R., *Intrinsic Motivation and Automatic Curricula via Asymmetric Self-Play*, arXiv:1703.05407 [cs], Apr. 2018. DOI: 10.48550/arXiv.1703.05407. [Online]. Available: http://arxiv.org/abs/1703.05407 (visited on 06/14/2023).

[110] Narvekar, S., Sinapov, J., and Stone, P., "Autonomous Task Sequencing for Customized Curriculum Design in Reinforcement Learning," pp. 2536–2542, 2017. DOI: https://doi.org/10.24963/ijcai.2017/353. [Online]. Available: https://www.ijcai.org/proceedings/2017/353 (visited on 06/14/2023).

[111] Google, *TensorBoard*. [Online]. Available: https://github.com/tensorflow/tensorboard (visited on 06/27/2023).

[112] ADAC, *Wallbox mit Lastmanagement*, Jun. 2020. [Online]. Available: https://www.adac.de/rund-ums-fahrzeug/elektromobilitaet/tests/wallbox-lastmanagement/ (visited on 01/12/2023).

[113] Chen, G. and Shi, X., *A Deep Reinforcement Learning-Based Charging Scheduling Approach with Augmented Lagrangian for Electric Vehicle*, arXiv:2209.09772 [cs] version: 1, Sep. 2022. DOI: 10.48550/arXiv.2209.09772. [Online]. Available: http://arxiv.org/abs/2209.09772 (visited on 07/24/2023).

[114] Behnel, S., Bradshaw, R., Citro, C., Dalcin, L., Seljebotn, D. S., and Smith, K., "Cython: The Best of Both Worlds," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 31–39, Mar. 2011, ISSN: 1521-9615. DOI: 10.1109/MCSE.2010.118. [Online]. Available: http://ieeexplore.ieee.org/document/5582062/ (visited on 08/10/2023).

[115] Dorokhova, M., Martinson, Y., Ballif, C., and Wyrsch, N., "Deep reinforcement learning control of electric vehicle charging in the presence of photovoltaic generation," *Applied Energy*, vol. 301, p. 117 504, Nov. 2021, ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2021.117504. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261921008874 (visited on 08/09/2023).

[116] Weiss, X., Xu, Q., and Nordström, L., "Energy Management of Smart Homes with Electric Vehicles Using Deep Reinforcement Learning," in *2022 24th European Conference on Power Electronics and Applications (EPE'22 ECCE Europe)*, Sep. 2022, pp. 1–9.

[117] Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M., *Dream to Control: Learning Behaviors by Latent Imagination*, arXiv:1912.01603 [cs], Mar. 2020. DOI: 10.48550/arXiv.1912.01603. [Online]. Available: http://arxiv.org/abs/1912.01603 (visited on 08/10/2023).

[118] Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., and Davidson, J., *Learning Latent Dynamics for Planning from Pixels*, arXiv:1811.04551 [cs, stat], Jun. 2019. DOI: 10.48550/arXiv.1811.04551. [Online]. Available: http://arxiv.org/abs/1811.04551 (visited on 08/10/2023).

[119] Ramesh, A. and Ravindran, B., *Physics-Informed Model-Based Reinforcement Learning*, arXiv:2212.02179 [cs], May 2023. DOI: 10.48550/arXiv.2212.02179. [Online]. Available: http://arxiv.org/abs/2212.02179 (visited on 08/10/2023).

[120] Janner, M., Fu, J., Zhang, M., and Levine, S., *When to Trust Your Model: Model-Based Policy Optimization*, arXiv:1906.08253 [cs, stat], Nov. 2021. DOI: 10.48550/arXiv.1906.08253. [Online]. Available: http://arxiv.org/abs/1906.08253 (visited on 08/10/2023).

[121] Zhao, P. and Liu, Y., "Physics Informed Deep Reinforcement Learning for Aircraft Conflict Resolution," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 8288–8301, Jul. 2022, Conference Name: IEEE Transactions on Intelligent Transportation Systems, ISSN: 1558-0016. DOI: 10.1109/TITS.2021.3077572.

[122] Liu, X.-Y. and Wang, J.-X., "Physics-informed Dyna-Style Model-Based Deep Reinforcement Learning for Dynamic Control," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 477, no. 2255, p. 20 210 618, Nov. 2021, arXiv:2108.00128 [cs, math], ISSN: 1364-5021, 1471-2946. DOI: 10.1098/rspa.2021.0618. [Online]. Available: http://arxiv.org/abs/2108.00128 (visited on 08/10/2023).

Appendix - Contents

Appendix A

FleetRL code structure

Environment class

The environment class forms the foundation of the framework, as it acts as the communication interface between the EV charging problem and the RL agent. The environment class of FleetRL is based on OpenAI's `gym.Env` class - the industry standard when it comes to implementing RL environments. When inheriting from `gym.Env`, three main functions must be implemented: `init`, `reset` and `step`.

In the `init` function, initial values, databases, and settings are loaded that are required when constructing a `gym.Env` instance. Most importantly, the `init` function defines the problem's observation and action space, which dictate whether the problem is continuous or discrete, and how many variables it includes.

```
def __init__(self):
```

- Load path and file names
- Set flags: e.g. include PV, building load, print errors
- Load config files: time config, EV config, penalty config
- Load modules: ev charging, time picker, observer, etc.
- Additional settings: target SoC, initial SoH, # of cars
- Set observation space: gym.spaces.Box()
- Set action space: gym.spaces.Box()

Figure A.0.1: Init function

The `reset` function resets the environment at the beginning of a new episode and returns the initial observation, and a dictionary with additional information.

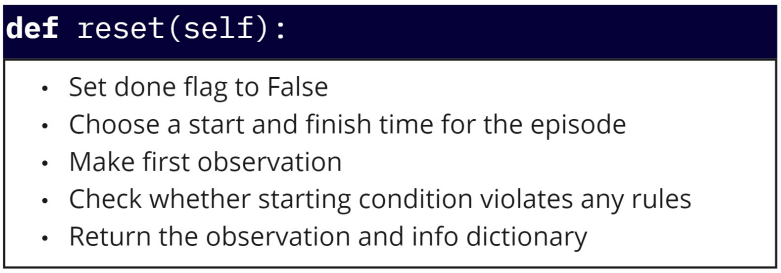


Figure A.0.2: Reset function

The `step` function requires the agent's actions as input, computes them and returns the new observation, the reward, two booleans and a dictionary with additional information. The two booleans indicate, whether an episode is done due to reaching a terminal state or a time limit, respectively.

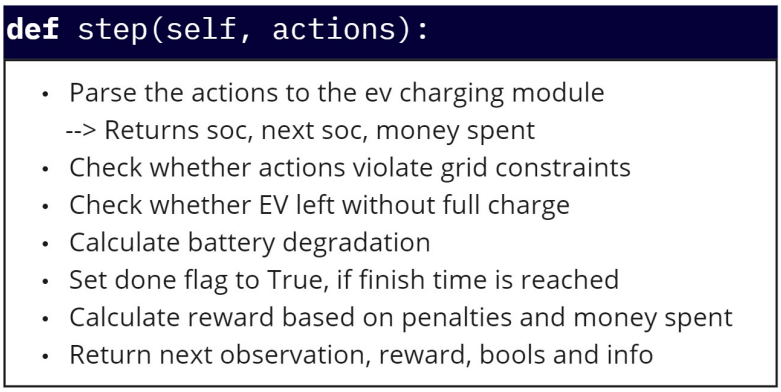


Figure A.0.3: Step function

Appendix B

5 EV Agent results

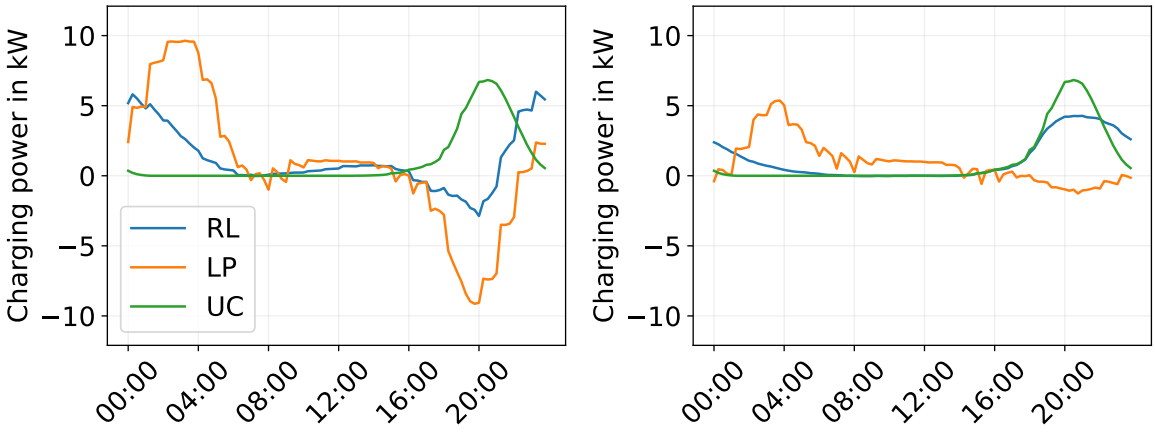


Figure B.0.1: LMD charging strategy

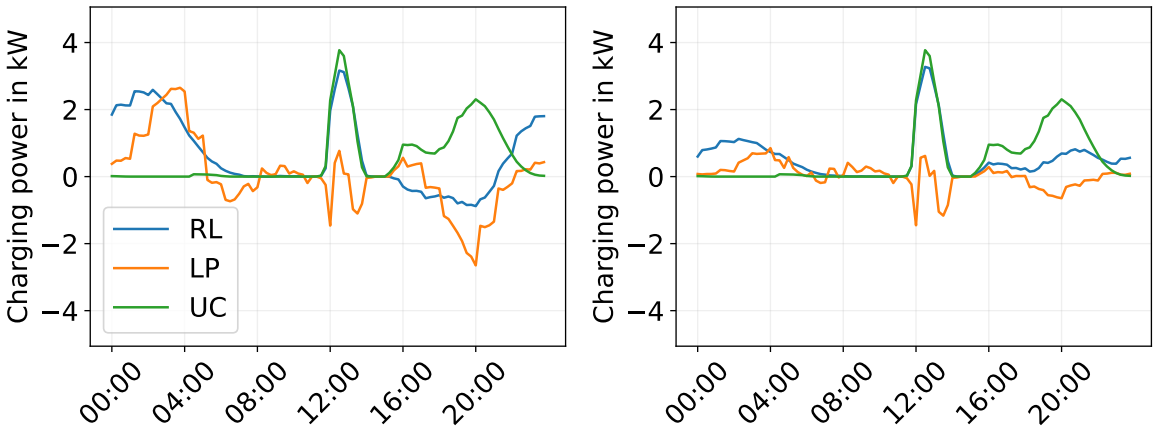


Figure B.0.2: CT charging strategy

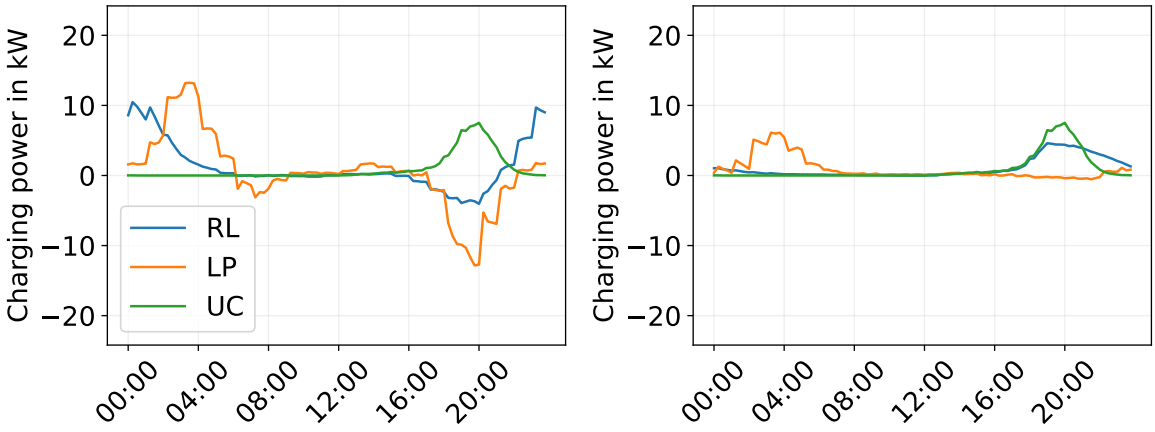


Figure B.0.3: UT charging strategy

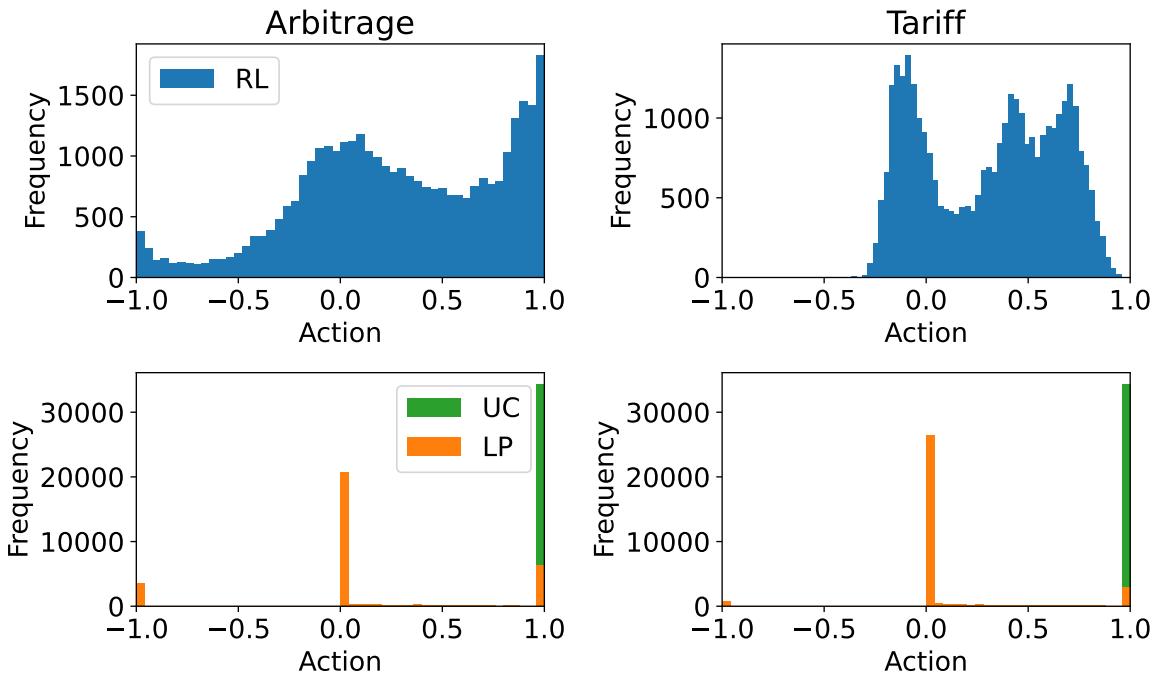


Figure B.0.4: Action distribution LMD

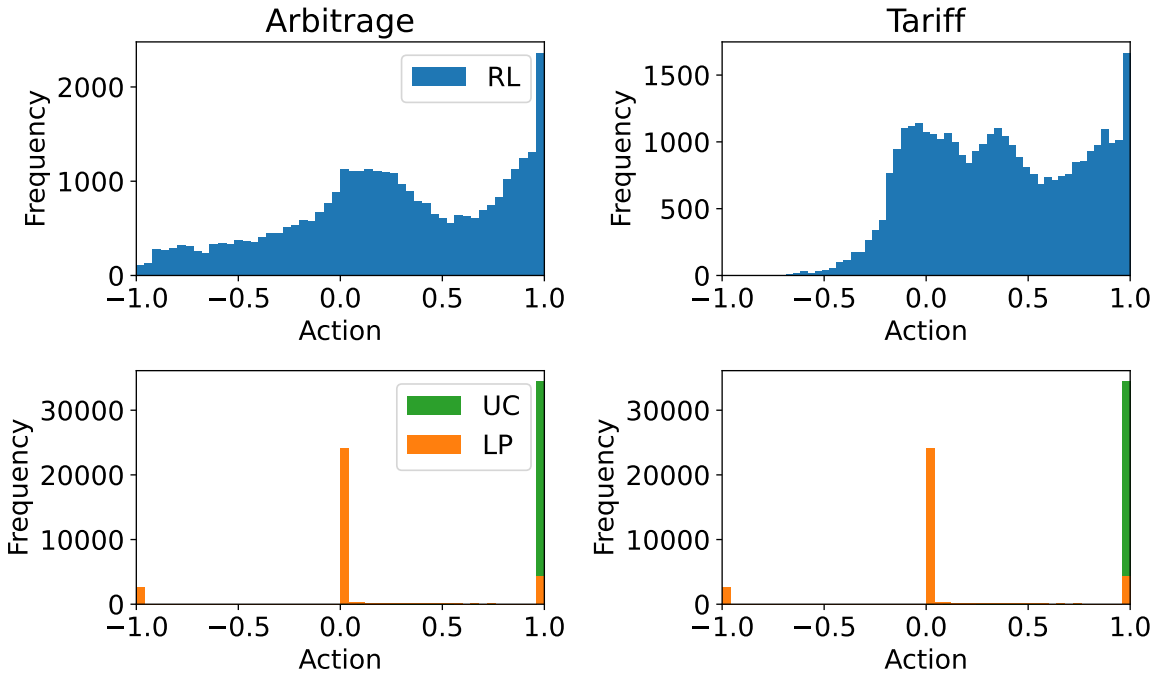


Figure B.0.5: Action distribution CT

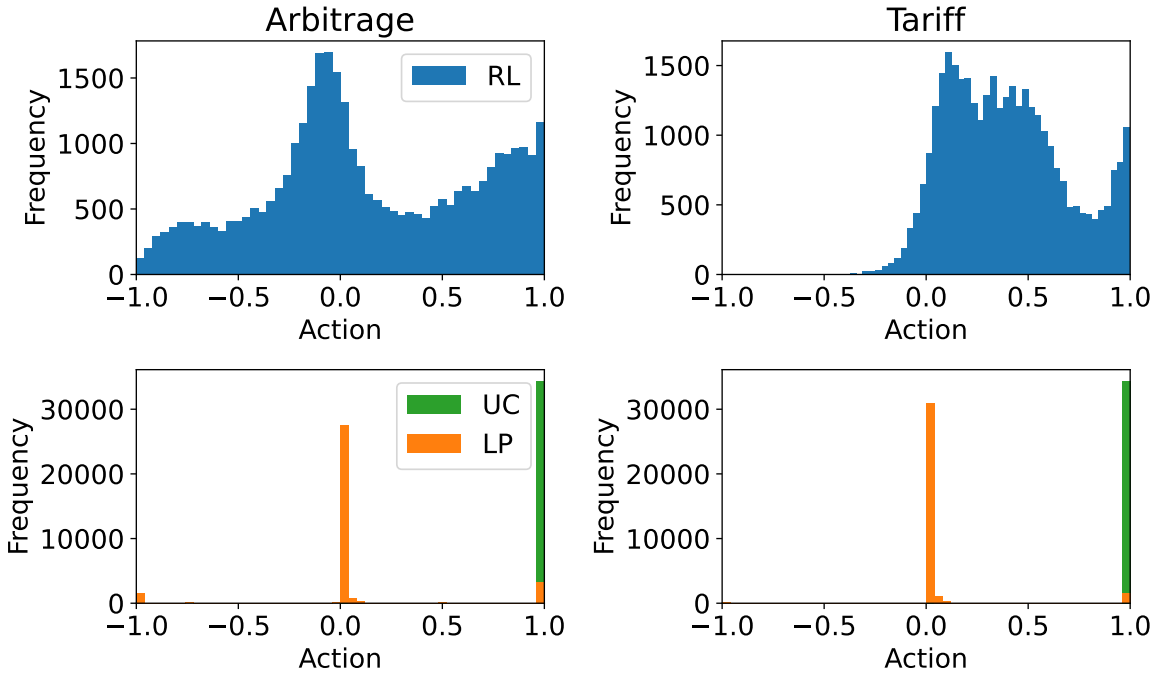
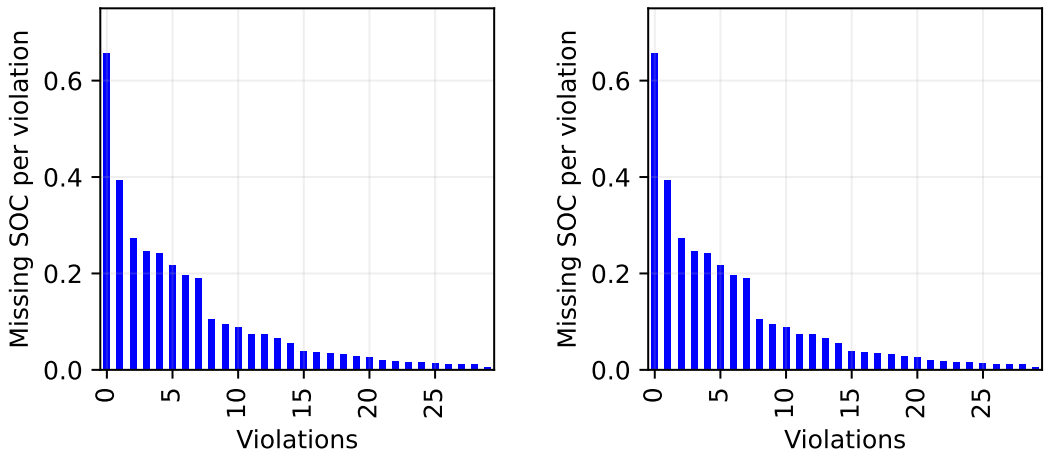
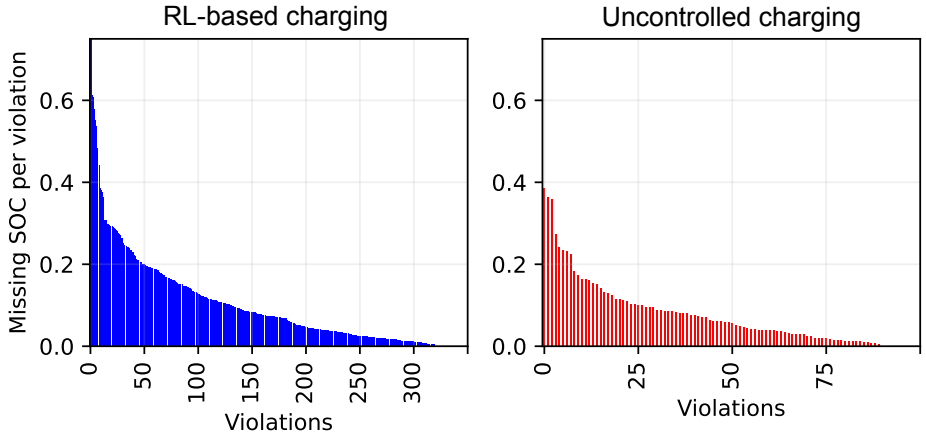


Figure B.0.6: Action distribution UT

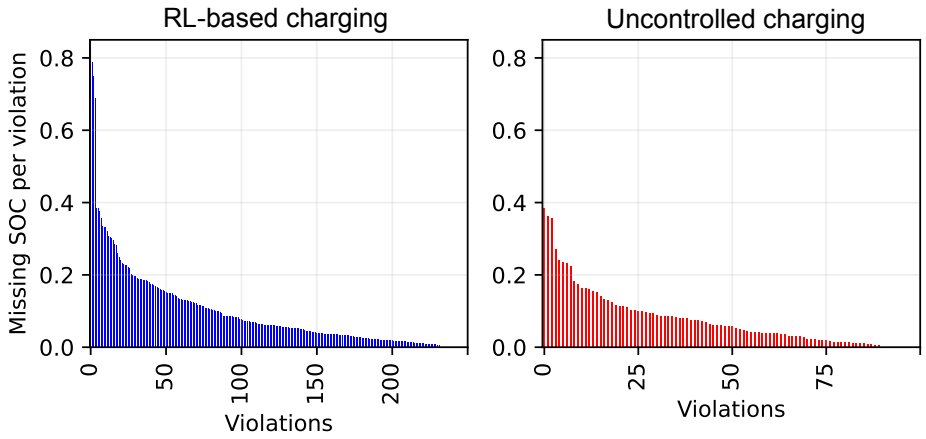


(a) LMD arbitrage (b) LMD tariff

Figure B.0.7: LMD - SOC violations for RL-based charging

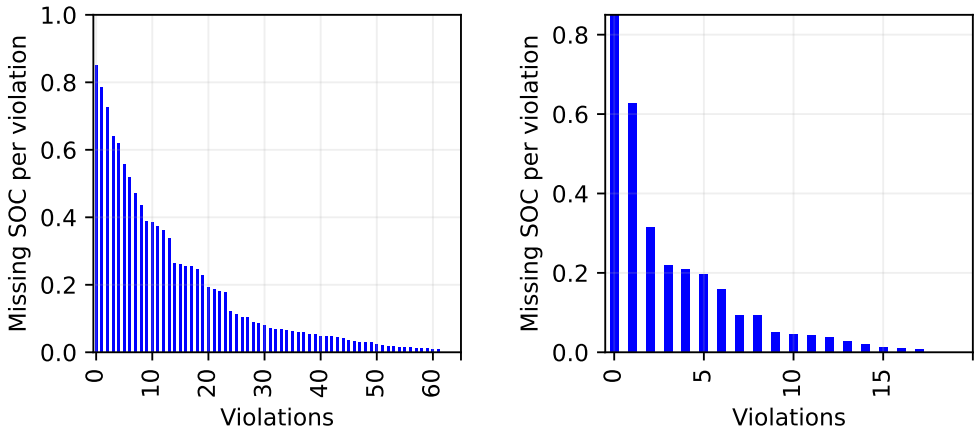


(a) CT arbitrage



(b) CT tariff

Figure B.0.8: CT - SOC violations for RL and dumb charging



(a) UT arbitrage (b) UT tariff

Figure B.0.9: Utility - SOC violations for RL-based charging

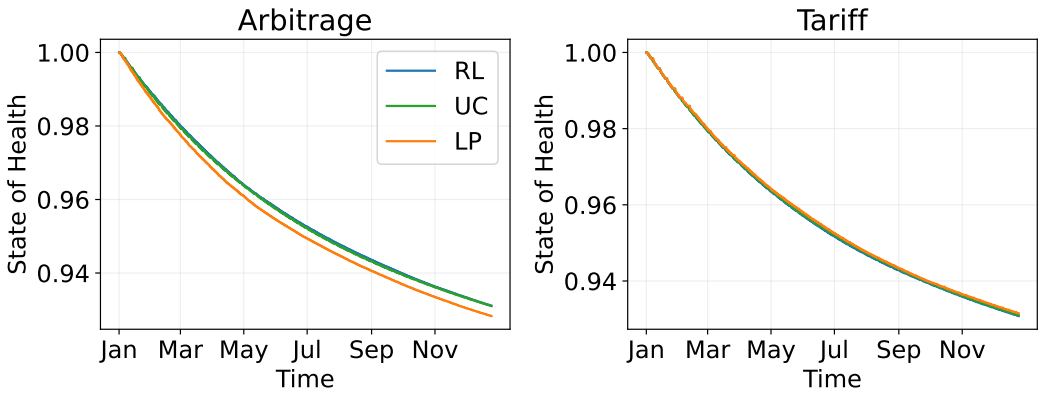


Figure B.0.10: Last-mile delivery - SOH for RL, LP and dumb charging

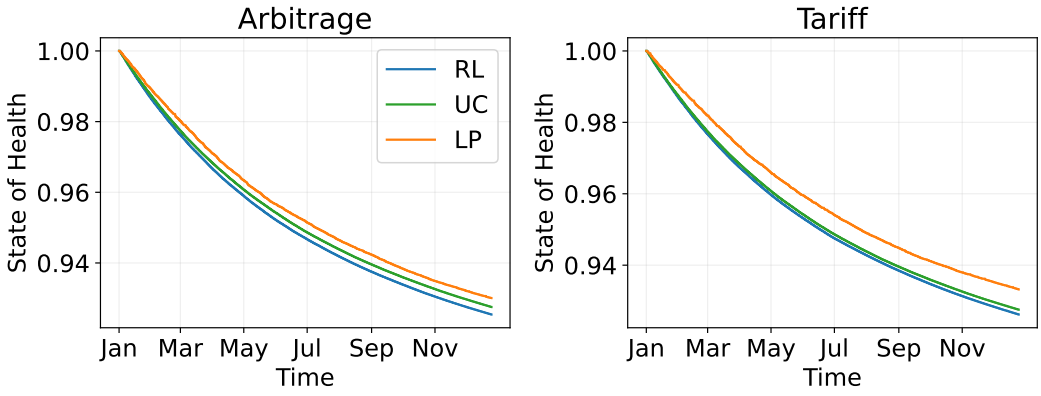


Figure B.0.11: Caretaker - SOH for RL, LP and dumb charging

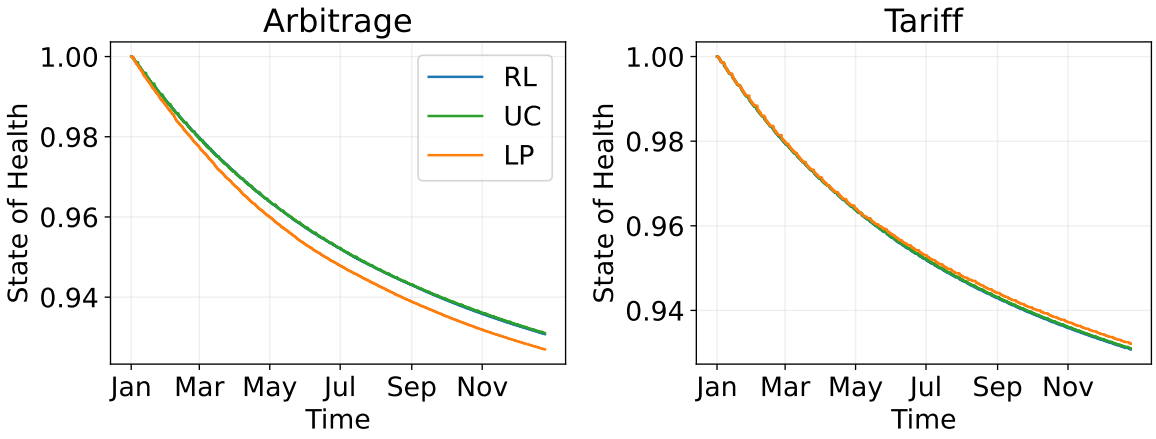


Figure B.0.12: Utility - SOH for RL, LP and dumb charging

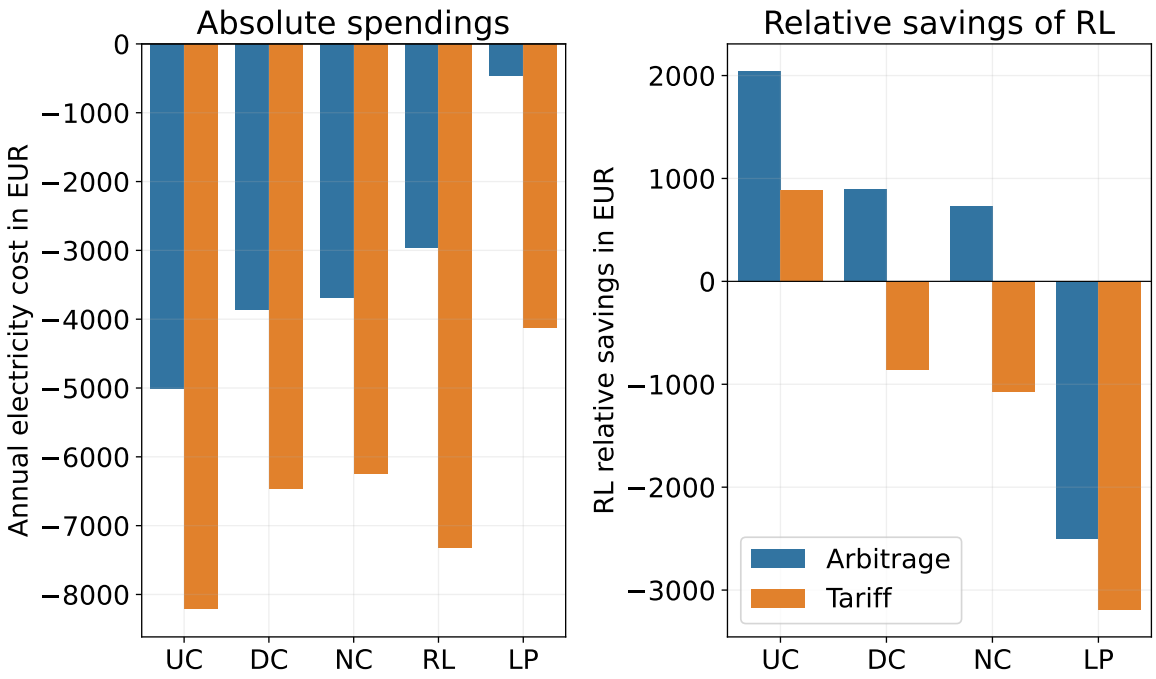


Figure B.0.13: LMD savings

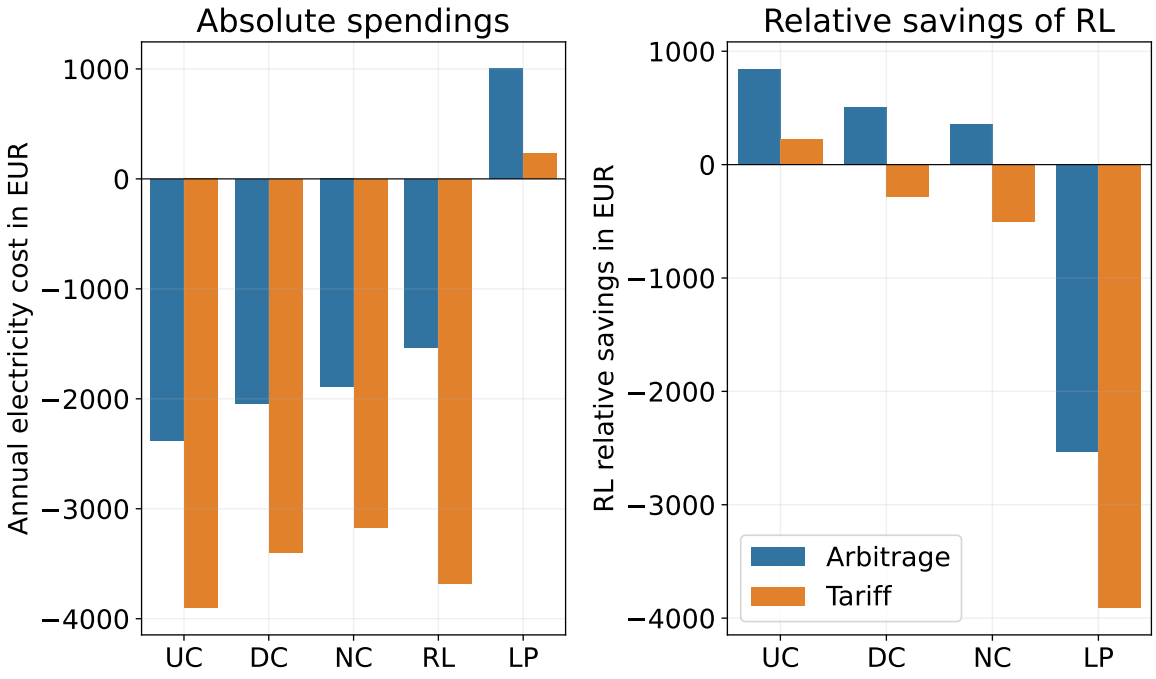


Figure B.0.14: CT savings

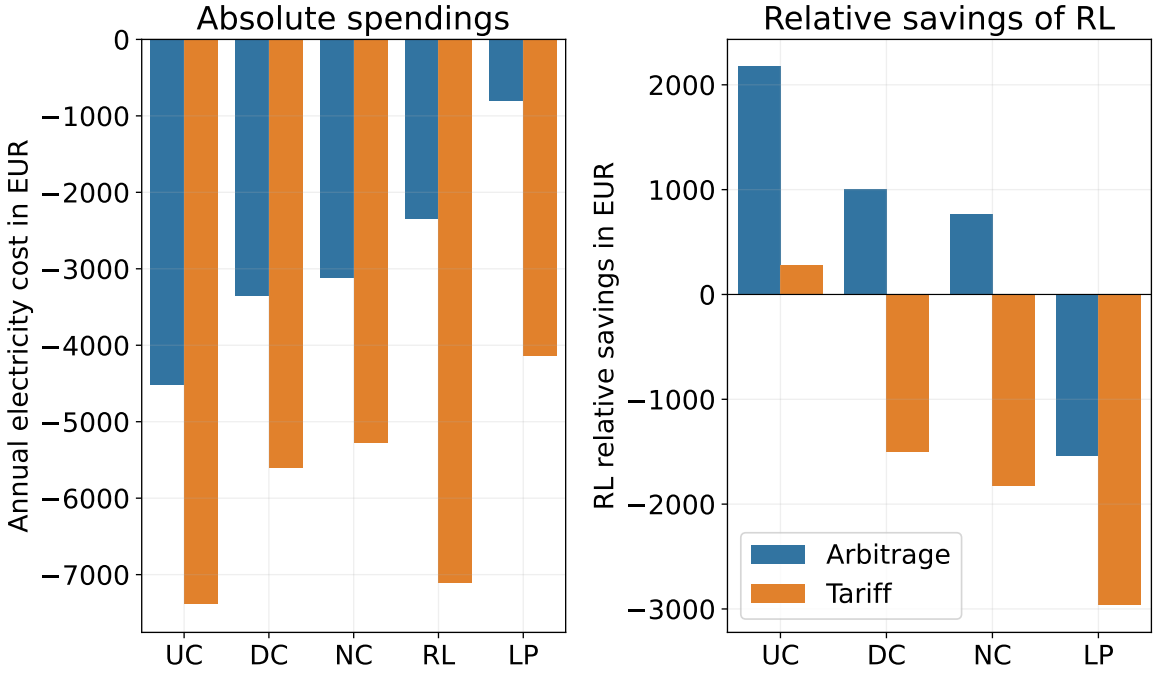


Figure B.0.15: UT savings