Universiteit Leiden

Master Computer Science

EVPriceNet: Dynamic Pricing for Electric Vehicle Charging Using Reinforcement Learning and Foundation Models

Name:	Donny Tsang
Student ID:	S2752883
Date:	31/12/2025
Specialisation:	Computer Science: Data Science
1st supervisor:	Marc Hilbert
2nd supervisor:	Joost Visser

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

The growing adoption of electric vehicles (EVs) in the Netherlands places increasing pressure on the electricity grid, particularly during peak demand hours. Dynamic pricing is frequently proposed as a mechanism to influence EV charging behaviour in order to maximize revenue and mitigate peak loads. However, existing studies on dynamic pricing often do not explicitly model network-level constraints and congestion effects. This thesis investigates which algorithmic approach is best suited for dynamic EV charging price optimization under semi-realistic operational constraints. Reinforcement learning methods, including tabular Q-learning, Deep Q-Networks (DQN), and Prioritized Dual Deep Deterministic Policy Gradient (PDDPG), were compared against transformer-based decision models. All methods generated a fixed 24-hour day-ahead price schedule and were trained to optimize a shared multi-objective reward function balancing revenue, congestion mitigation, and operational stability. The evaluation was conducted using *EVPriceNet*, a simulation environment calibrated on historical Dutch EV charging demand and electricity price data, which captures stochastic arrivals, local competition between charging stations, and congestion-sensitive constraints.

Contents

List of Abbreviations

EV Electric Vehicle

BEV Battery Electric Vehicle

PHEV Plug-in Hybrid Electric Vehicle

RL Reinforcement Learning

POMDP Partially Observable Markov Decision Process

DQN Deep Q-Network

DDPG Deep Deterministic Policy Gradient

PDDPG Prioritized Deep Deterministic Policy Gradient

PER Prioritized Experience Replay

DT decision transformer

ODT online decision transformer

FM Foundation Model

TOU Time-of-Use

CPP Critical Peak Pricing

VPP Variable Peak Pricing

RTP Real-Time Pricing

PTR Peak Time Rebates

PED Price Elasticity of Demand

ENTSO-E European Network of Transmission System Operators for Electricity

Nomenclature

α	Learning rate
ΔP	Change in price relative to the baseline price
ΔQ	Change in quantity demanded after a price change
δ_{comp}	Indifference threshold for switching to a competitor station [EUR/kWh]
δ_{delay}	Indifference threshold for delaying a charging session [EUR/kWh]
γ	Discount factor
κ	Smoothing parameter for combining daily and global arrival profiles
λ_t^{day}	Hourly arrival rate for a specific calendar day
$\lambda_t^{\text{global}}$	Average hourly arrival rate over all days
λ_t^{total}	Total expected EV arrival rate across the network at hour t
λ_t	Expected EV arrival rate per station at hour t
$\mathcal{A}$	Action space
$\mathcal{E}$	Set of proximity edges between stations
$\mathcal{P}$	State transition dynamics
$\mathcal{R}$	Reward function
$\mathcal{S}$	State space
$\mathcal{V}$	Set of charging stations (nodes)
π	Policy mapping states to actions
π_t^{wh}	Wholesale electricity price at hour t [EUR/kWh]
c_t	Congestion indicator at hour t
$E_t^{\text{competitor}}$	Energy delivered by competitor stations at hour t [kWh]
$E_t^{\text{delivered}}$	Total energy delivered by the agent at hour t [kWh]
$E_s^{\text{remaining}}$	Remaining energy demand of session s at departure [kWh]
$E_t^{\text{shortfall}}$	Total unmet energy demand at hour t [kWh]
$G = (\mathcal{V}, \mathcal{E})$	Charging network graph
k_{comp}	Price sensitivity parameter for choosing between agent and competitor
k_{delay}	Price sensitivity parameter for delaying a charging session

M Scaling factor calculation month

N_t Number of EV arrivals during hour t

P Baseline price of the good before a price change

p_t Retail charging price at hour t [EUR/kWh]

$p_{\max}$ Maximum allowed charging price [EUR/kWh]

Q Baseline quantity demanded

$Q(s, a)$ Action-value function

S_t^{lost} Number of lost charging sessions at hour t

s_M Scaling factor

t Hourly time index within a day

t_{global} Global timestep across days

w_{comp} Competitor penalty weight

w_{cong} Congestion penalty weight

w_{lost} Lost session penalty weight

w_{rev} Revenue weight

w_{short} Shortfall penalty weight

w_{vol} Price volatility penalty weight

1 Introduction

The growing adoption of electric vehicles (EVs) in the Netherlands poses new challenges for the electricity grid, including net congestion during peak demand periods. As the share of EVs continues to rise, the pressure on the local electricity grid grows accordingly [43]. This trend, called grid congestion or net congestion, creates situations where demand for electricity can exceed the capacity of the grid. In turn, this could result in power outages and a weakened reliability of the power grid [11] and is becoming an increasingly urgent issue for grid operators. The expansion of the grid infrastructure cannot keep up with the increasing demand for electricity in the Netherlands, which can become a bottleneck for transitions to net-zero emissions [23]. Therefore, addressing this challenge requires smarter utilization of existing grid capacity rather than solely relying on infrastructure expansion.

A development that could help reduce grid congestion is the application of dynamic pricing models [13]. Citing from Araman and Caldentey [6]: "Consider a seller who is endowed with a fixed number of units of a product that she can sell to a price-sensitive and stochastically arriving stream of consumers during a given selling season. The seller's problem is to dynamically adjust the product's price to maximize the revenues she can collect if no replenishment is possible during the sales horizon." Dynamic pricing refers to the practice of adjusting prices over time depending on different factors. An example is the hotel industry, where dynamic pricing methods are already applied [1]. Hotels can adjust room prices depending on the amount of rooms occupied already or even the price of the rooms of a nearby competitor. It is in the interest of the hotel to maximize its revenue by changing the prices based on these factors. When there is a scarce resource to sell, dynamic pricing techniques can be used to maximize the revenue made. Therefore, dynamic pricing is originally a technique used for revenue maximization [16]. In the context of EV charging, the scarce product would be electricity, a resource that is and should be limited by the capacity of the grid. The EV drivers, or the consumers of the electricity, will arrive stochastically and are assumed to be partially price-sensitive. Dynamic pricing methods can therefore be used for maximizing revenue for the charging station operator. However, it can be used to contribute to reducing net capacity problems based on the proposed charging price as well. It has been seen in the electricity usage of households that a reduction between 3-6% is possible using dynamic pricing [19]. With critical peak tariffs even leading to a reduction of demand between 13-20%.

Using an algorithmic approach to dynamic pricing has been observed to be more effective in maximizing revenue when compared to short-term pricing decision-making [8]. These methods should also take into consideration different inputs like market price, driver behaviour and demand forecast [17]. To use dynamic pricing optimally, there is a strong need for a method that can learn from historical patterns and adapt dynamically to changing conditions. A key part of using an algorithmic approach to dynamic pricing is the model. Traditional regression models have the advantage of transparency and easy deployment, but may struggle to capture more complex input features. Recent advances in machine learning introduce foundation models [31]. These are transformer-based architectures and have the ability to capture more complex behavioural patterns. This thesis will further explore which type of model is best suited for predicting the impact of EV driver charging behaviour under fluctuating prices.

To evaluate and compare different dynamic pricing approaches, a simulation environment is required. Therefore, this project will develop an extended version of the open-source CharGym environment [25], originally designed for applying reinforcement learning methods to control the charge and discharge of different EV charging stations. This modified environment,

called EVPriceNet, will simulate multiple EV charging stations, price-sensitive EV drivers and competitors using differing dynamic pricing strategies. EVPriceNet will integrate real-world electricity pricing data and charger profiles. This environment should provide a semi-realistic setting for testing differing dynamic pricing strategies.

1.1 Research Questions

Based on the identified challenges and objectives, this thesis will address the following research questions:

1. RQ1: Is the custom environment stable, reproducible, and representative enough to support training and comparison of different algorithmic methods?
2. RQ2: To what extent do the selected reinforcement learning methods (Q-learning, DQN, DDPG and PDDPG) differ in their ability to optimize dynamic EV charging prices under semi-realistic constraints within the simulated environment?
3. RQ3: To what extent do transformer-based models (decision transformers and time-series foundation models) compare to the performance of reinforcement learning methods when optimizing dynamic EV charging prices under semi-realistic constraints within the simulated environment?

2 Background

Dynamic pricing can be used to influence consumer behaviour to reduce peak hour demand by providing a financial incentive [19]. However, designing effective pricing strategies requires models that can take incorporate multiple input factors [17]. Reinforcement learning provides a framework that allows pricing policies to be learned through interaction with a simulated environment. Some transformer-based models can be trained from historical data, while others can be used without online interaction, enabling zero-shot or offline decision-making [31]. This section introduces fundamental concepts for these approaches and serves as a introduction for the methods explored in this project.

2.1 Dynamic pricing for electricity and EVs

Dynamic pricing for electricity is based around tariffs, which changes the price of electricity over time based on changes in supply–demand conditions, grid capacity constraints, or market signals. Dutta and Mitra [17] identify multiple different pricing policies, which are cited below.

- *Flat tariffs: The price remains static even though power demand changes. Consumers under such a scheme do not face the changing costs of power supply with a change in aggregate demand. Thus, consumers have no financial incentive to reschedule their energy usage. They do not face any risk of high-value electricity bills for any unavoidable or unplanned electricity consumption. Hence, this scheme is often used as a welfare pricing scheme.*
- *Block Rate tariffs: This scheme differentiates between customers based on the quantity of electricity consumption. The scheme consists of multiple tiers characterized by the*

amount of consumption. Inclining rate schemes increase the per-unit rate with increasing consumption and declining schemes do the opposite.

- *Seasonal tariffs: These schemes observe different rates in different seasons to match the varying demand levels between seasons. Energy is charged at a higher rate during high-demand seasons and the price lowers during low-demand seasons.*
- *Time-of-use (TOU) tariff: These are pre-declared tariffs varying during the different times of the day, that is, high during peak hours and low during off-peak hours. Such schemes can stay effective for short or long terms. This is also known as time-of-day (TOD) tariff.*
- *Superpeak TOU: It is similar to TOU, but the peak window is shorter in duration (about four hours) so as to give a stronger price signal*
- *Critical peak pricing (CPP): This is a dynamic pricing scheme in which prices are high during a few peak hours of the day and discounted during the rest of the day. The peak price remains same for all days. It gives a very strong price signal and enhances the reduction of excessive peak load.*
- *Variable peak pricing (VPP): This is quite similar to CPP with the only difference that the peak prices vary from day to day. The consumers are informed about such peak prices beforehand.*
- *Real-time pricing (RTP): This is the purest form of dynamic pricing and the scheme with the maximum uncertainty or risk for the consumers. Here the prices change at regular intervals of 1 h or a few minutes. The change in the price in such small intervals increases the efficiency of the pricing scheme in reflecting the actual costs of supply, but such schemes require advanced technology to communicate and manage these frequent Journal of the Operational Research Society changes. Retail electricity markets may find it difficult to practice this scheme due to the high rate of data collection and transfer.*
- *Peak time rebates (PTR): These rebates are just the opposite of CPP schemes. Rebates are provided for consuming below a certain predetermined level during peak hours and can be redeemed at a later time.*

The effectiveness of using these differing dynamic pricing schemas is dependent on the concept of price elasticity of demand (PED), which measures the change in consumer behaviour based on a change in price. Equation 1 can be used to calculate PED with Q being the quantity of the demanded good, ΔQ being the difference in demand after a price change, P being the reference price and ΔP being the difference in price.

$$PED = \frac{\Delta Q/Q}{\Delta P/P} \tag{1}$$

The PED being negative is a usual outcome in the case of electricity. This implies that the demand of electricity does decrease when the price increases, although the magnitude of the response is often considered small. When $|PED|$ is smaller than 1, the good is called price inelastic, which is usual for electricity. General household electricity demand often shows relatively low short-term elasticity. For example, Lijesen [32] report elasticity values ranging from -0.05 to -0.6. These lower values reflect the limited flexibility of many household electricity uses, such as lighting or essential appliances.

EV charging in particular may show higher elasticity due to its more flexible nature. Unlike many household electricity usage, EV charging can be postponed, especially in public or workplace settings. Another factor is that users may choose between different charging locations or providers for every single session, while household electricity is usually not as flexible. Kuang et al. [29] estimate the price elasticity of EV charging demand in Shenzhen, China to lie between -0.7236 and -0.7654, with an average of -0.7581. These values suggest a stronger behavioural response compared to general electricity consumption. For congestion management purposes, relatively small demand shifts during peak periods may already be sufficient to help grid reduce grid congestion. Borenstein [10] estimate that bringing a third of the electricity consumers from flat tariffs to RTP would reduce peaking capacity requirements by 44%.
Therefore, even when demand is price inelastic, dynamic pricing can still result in meaningful behavioural adjustments. Empirical studies on TOU and CPP in electricity markets show measurable reductions in peak demand, typically in the range of 3–6% under TOU schemes and up to 13–20% under CPP [20, 2]. Several studies report that EV users demonstrate temporal flexibility and responsiveness to price-based incentives, particularly when charging is not immediately required [40, 45]. Furthermore, the presence of local competition between charging stations may strengthen price responsiveness, as users can switch providers when price differences become larger.

2.2 Reinforcement learning fundamentals

Reinforcement Learning (RL) is a field within machine learning in which an agent learns to make decisions through interactions with an environment [24]. At each timestep, the agent observes the current state, selects an action and receives a reward dependent on the quality of that action. The objective of this agent is to learn a policy π that maximizes the expected cumulative reward over time. RL problems are often formalized as Markov Decision Processes (MDPs) [44], defined by $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ the action space, P the transition dynamics, R the reward function, and $\gamma \in [0, 1]$ the discount factor. Oftentimes, the environment can be only partially observable or stochastic, making exact modelling of P impractical.
RL algorithms can vary in their methods to optimize a policy. The following subsections introduce different RL methods that are used in this project.

2.2.1 Q-learning

Q-learning is an RL algorithm that tries to learn the optimal action-value function $Q(s, a)$ [53]. This function is the expected cumulative reward obtained by taking action a in state s. In tabular Q-learning, the agent stores estimated Q-values for each state-action combination in a table, which is called the Q-table. During interactions with the environment, the agent updates these estimates using this update rule:

$$Q(s, a) \leftarrow Q(s, a) + \alpha\Big(r + \gamma \max_{a'} Q(s', a') - Q(s, a)\Big) \tag{2}$$

with α being the learning rate, r the observed reward for taking action a in state s, s' the next state and a' being the candidate next action in the next state. By applying this update function during the interaction with the environment, the Q-learning agent can converge to an optimal action-value function. However, a limitation with Q-learning is that it does not work well in environments with large or continuous state-action spaces. As the number of possible states

grows, the Q-table will become increasingly sparse, which will lead to slow learning and poor generalization between similar states. In these cases, the agent might frequently find states previously unseen and will have to take uninformed actions. Q-learning can, however, still serve as a useful baseline for comparison.

2.2.2 DQN

Deep Q-Networks (DQN) extend Q-learning by approximating the action-value function using a deep neural network [38]. Instead of storing Q-values for each state-action pair, the network learns a function $Q_\theta(s, a)$ that can generalize across similar states. The DQN is trained by minimizing the temporal difference error between predicted Q-values and target values derived from the Bellman optimality equation, seen in Equation 3.

$$Q(s, a) = r + \gamma \max_{a'} Q(s', a') \tag{3}$$

where r is the immediate reward, $\gamma \in [0, 1]$ is the discount factor, s' is the next state, and a' denotes the next action. In practice, DQN approximates $Q(s, a)$ by minimizing the discrepancy between both sides of the Bellman equation using samples collected from interaction with the environment.
Direct training of such a network is often unstable due to correlations in sequential data and changing targets. To minimize this problem, the DQN introduces two stabilization techniques. Experience replay stores observed transitions (s, a, r, s') in a replay buffer and samples mini-batches during training. This breaks temporal correlations and allows transitions to be reused.
Target networks stabilize learning by maintaining a set of parameters for calculating target Q-values. The target network is updated less frequently than the main network and reduces feedback loops where the network attempts to find rapidly changing predictions. Nevertheless, the DQN remains a commonly used baseline for high-dimensional decision-making problems and is therefore included as a reference method in this work.

2.2.3 DDPG

The Deep Deterministic Policy Gradient (DDPG) is an actor-critic reinforcement learning method designed for environments with continuous action spaces [33]. Unlike Q-learning and DQN, which select actions by maximizing an action-value function, the DDPG directly learns a deterministic policy that maps states to continuous actions.
A DDPG consists of two neural networks: an actor network and a critic network. The actor receives the state as an input and outputs the action in the continuous space. The critic estimates the action-value function $Q(s, a)$ for the given state-action pair and is then trained to minimize temporal difference error, similar to the DQN. The actor is trained by following the gradient of the critic. This allows the policy to be improved in a continuous action space. The DDPG also uses experience replay and target networks like the DQN to stabilize the training. By combining these techniques, the DDPG is a method well suited for continuous action spaces.

2.2.4 PDDPG

Prioritized Deep Deterministic Policy Gradient (PDDPG) combines the DDPG with prioritized experience replay (PER) [45]. Instead of sampling transitions uniformly from the replay buffer,

PER assigns higher sampling probabilities to transitions with higher temporal difference errors. As a result, the agent focuses more frequently on experiences that are expected to yield greater learning progress. This enhancement is beneficial in environments where rewards are noisy or sparse [45]. By emphasizing these transitions during training, PDDPG can converge more reliably than standard DDPG.
Qiu et al. [45] use this method in a similar space when trying to maximize revenue for charging station operators using dynamic pricing. The PDDPG achieves the highest average profit there when compared to Q-learning, DQN and DDPG.

2.3 Transformer model fundamentals

Transformer models are neural network architectures originally introduced for sequence modelling tasks in natural language processing (NLP) [50]. Transformers process entire sequences in parallel using self-attention mechanisms, allowing them to capture long-range dependencies without relying on sequential state updates.
In recent years, transformer-based models have been successfully adapted to time-series data. In this setting, sequences consist of temporal observations rather than tokens. Attention mechanisms allow the model to capture both short-term and long-term seasonal patterns. Transformers can be applied to decision-making problems by modelling observations, actions and outcomes. In some settings, transformer-based models can be used to generate actions by conditioning on historical trajectories. This allows them to learn directly from offline data and avoids the instability and exploration challenges associated with reinforcement learning. The following sections will discuss the fundamentals of decision transformers and foundation models.

2.3.1 Decision transformers

Traditional decision transformers (DTs) take a fundamentally different approach to decision-making when compared to RL. Instead of trying to optimize a policy through online interactions with an environment, DTs model decision-making as a sequence modelling problem using a transformer architecture [12] where actions are generated by conditioning on trajectories of past behaviour. In the classical setting, DTs do not interact with the environment during training and are trained offline.
A classical decision transformer uses a trajectory, which consists of a sequence of tuples containing a return-to-go, state observations and actions. The return-to-go represents the actual realized cumulative reward given in training data for a given timestamp until the end of the episode. By conditioning on a desired return-to-go, the model learns to generate actions that will lead to similar outcomes as in the training data. The return-to-go is therefore different than an expected return as in RL and guides the model toward higher-performing behaviour.
Recent work extends the decision transformer framework to online and interactive settings [57]. The model is no longer trained on static offline datasets but can collect experience through environment interaction, just like RL. These online decision transformers (ODTs) update their trajectories incrementally with newly observed transitions. This allows a DT to fine-tune and update its own policy, similar to RL.
ODTs therefore follow a hybrid training method. While the model does not explicitly optimize a policy via value functions or policy gradients, policy improvement can emerge implicitly as higher-performing trajectories generated during interaction become increasingly prevalent in the

training data. The return-to-go remains part of the model input, but it now reflects realized outcomes achieved by the current policy rather than a fixed historical dataset.
For the problem of dynamic pricing, online decision transformers are not yet widely studied to the best of the author's knowledge. Similar to RL-based methods, the agent interacts with the environment and adapts its behaviour based on observed outcomes. At the same time, the learning objective remains a sequence prediction problem, allowing the use of transformer architectures that can condition on long temporal contexts and rich state representations. In this sense, online decision transformers can occupy a position between classical model-free reinforcement learning methods and fully pre-trained foundation models.

2.3.2 Time-Series Foundation models

Foundation Models are large-scale neural networks trained on diverse datasets with the goal of creating a model that can be used generally and transferred to different tasks [31]. Unlike traditional task-specific models, foundation models are designed to be reused and adapted with minimal additional training, enabling transfer to new problems and application settings.
In the time-series domain, foundation models are trained on large collections of temporal data across multiple domains and regions. Time-series foundation models are pre-trained and have learned generic temporal patterns across seasonality, trends and differing domains. Once trained, these models can be applied to zero-shot new tasks, often without needing more fine-tuning [3, 14].
An important characteristic of time-series foundation models is that they are trained to predict future observations based solely on historical context. They do not rely on explicit knowledge of the underlying data and also do not require interaction with the environment. They learn probabilistic mapping from past observations and can be applied in multiple domains so long as the input data can be represented as a time-series.
Recent work has explored the use of time-series foundation models in energy-related domains, including electricity price forecasting and load prediction [55]. However, it is important to note that foundation models are not learning methods like reinforcement learning. They do not optimise objectives or policies as their strength lies in forecasting and pattern continuation. They are fully trained models and, as previously described, have no need for interaction with the environment. When used in an operational setting, foundation models are strong as a general-purpose temporal predictor that can support optimisation-based or learning-based methods.
Foundation models offer several potential advantages in the context of EV charging price setting. First, they are trained on large and diverse time-series datasets and therefore capture general temporal patterns such as seasonality. A foundation model that has been pre-trained on diverse time-series data may therefore already possess useful representations of such dynamics, even without domain-specific fine-tuning. Second, in a zero-shot context these models can generate forecasts without environment interaction or task-specific training data. Reinforcement learning methods require simulation and exploration to learn a policy, while foundation models can be used without observations from an environment. This could reduce training complexity and computational costs. Finally, foundation models can be very useful in settings where data is limited as they already have been trained on during pre-training.

3 Related work

Dynamic pricing in the context of electric vehicle charging is a novel research area. Although dynamic pricing is widely studied in broader electricity markets, its application to EV charging remains limited. This section reviews existing work on dynamic pricing, RL approaches, and data-driven modelling for energy systems.

3.1 Dynamic pricing strategies for EV charging

Chen and Folly [13] show a comprehensive review of the current state of artificial intelligence in EV charging for charging and discharge scheduling and also dynamic pricing. The paper identifies three categories of artificial intelligence application for EV charging, which are forecasting, scheduling, and pricing. The authors recognise that many companies used Time-of-use (TOU) tariffs to encourage users to shift their demand from peak periods to non-peak periods, which can result in lowered peak demand [47]. However, a new and potentially more significant peak might form during originally non-peak hours due to the amount of shifted demand of EVs [9]. Real-time pricing (RTP) could potentially overcome the challenges created by TOU tariffs due to the frequency it updates [9]. Hourly-based RTP has also been shown to benefit the EV drivers [56]. However, hourly RTP still cannot completely reflect the conditions of a power grid, which are instant. Instantaneous RTP has been proposed to more closely reflect the actual conditions of the power system and electricity market [9]. Therefore, it is important to consider the updating frequency when designing an RTP pricing scheme.
Although RTP is effectively used in other fields, there is still limited research on its application for EV charging [13]. Usually in this sector, power system operators use real-time power grid conditions, such as load demand, electricity price, and renewable energy generation, to generate prices to change charging and discharging behaviours [13]. Possible goals of RTP strategy can be to maximize the welfare of the charging system [35], power grid stability [7], reduce charging wait time [54], minimize EV charging cost [26] or maximize revenue of the charging station operator [30]. Kim et al. [26] proposed a RL-based approach for RTP that considers the load demand of EV drivers and electricity cost. Their simulation resulted in a RL-based dynamic prices that achieved higher and more long-term performance than an approach that only focuses on immediate system performance.

3.2 Models in literature

RL is a good baseline method for creating dynamic prices and has been used in previous papers [26, 39, 36]. Luo, Huang, and Gupta [36] use day-ahead wholesale electricity price, EV charging demand, and renewable energy generation to calculate dynamic prices. Moghaddam et al. [39] propose an online RL method for RTP for EV charging stations. A recent study, Qiu et al. [45], applies RL and compares Q-learning with a DQN, DDPG and PDDPG. In their simulation, the PDDPG achieves the best performance compared to the other methods.
Next to RL methods, recent developments in time-series foundation models introduce new possibilities for modelling complex temporal patterns that influence electricity pricing. Recent work done by Yu et al. [55] introduced a transformer-based foundation model for electricity price forecasting across multiple regions. Other relevant and recent time-series foundation model like TimesFM [14] or Chronos[3], developed by Google Research and Amazon respectively, show that foundation models are capable of capturing spatial and temporal input, which are also

important for the pricing of electricity. These aforementioned models are used for forecasting and are not policy-based however. At the time of writing, no foundation model has been applied yet to generate dynamic prices.
Among these transformer-based models, there are the decision transformers [12]. These interpret decision-making as a sequence modelling problem and have the ability to process multivariate inputs to learn policies from offline data like RL methods. Therefore, DTs can serve as a bridge between RL methods and pure time-series foundation models such as TimesFM or Chronos [14, 3].
Using reinforcement learning approaches can outperform static or rule-based pricing strategies. However, they require online interaction and are sensitive to hyperparameters tuning [24]. Transformer based models such as decision transformers and time-series foundation models offer the ability to learn pricing behaviour directly from historical trajectories. Despite their success, these models have not been systematically evaluated for EV charging price optimisation. This thesis addresses this gap by comparing reinforcement learning agents with transformer-based models in a unified simulation environment.

3.3 Testing environments

Developing and benchmarking dynamic pricing algorithms for EV Charging requires a simulation environment that can simulate the entire charging station operation and EV driver behaviour. While several EV-focused simulators exist, most are designed for charging control rather than pricing strategies. Karatzinis et al. [25] introduces CharGym, which is an open-source gym environment that simulates grid-connected EV charging stations with a fixed number of charging spots and stochastic EV arrivals and departures. CharGym models the state of charge of the batteries in EVs and limits charging by the grid constraints. The actions taken by the policies correspond to charging and discharging and cannot directly be used for testing dynamic pricing policies.
Despite the difference in objective, the environment developed in this thesis is structurally inspired by CharGym. Several modelling choices are adopted or extended, such as the daily episodes with hourly timesteps, stochastic EV arrivals, departures and session-level representations, with remaining energy demand and departure deadlines. Similar to CharGym, physical capacity constraints are enforced at the site level and penalties are applied when EVs depart without receiving their required energy.
However, CharGym models charging and discharging as actions, while this thesis defines the action space as to represent day-ahead pricing. Charging power is not directed by the agent. As a result, the proposed environment does not focus on energy control, but on behavioural decision-making. It does retain the realistic charging dynamic introduced by CharGym however.
Other environments are not EV-specific. For example, CityLearn is an open-source gym environment for training agents in realistic building and grid systems [51]. In the EV domain, Orfanoudakis et al. [42] introduce EV2Gym, which is a vehicle-to-grid simulator and can also be used to train RL agents. This paper also criticizes the simplified nature of CharGym [25].
Mahmud et al. [37] create an OpenAI Gym environment for real-time EV charging price optimisation, in which an agent selects hourly prices and receives rewards based on revenue and load-shifting objectives. Their framework tightly integrates demand forecasting with deep reinforcement learning and demonstrates that dynamic pricing strategies can outperform static pricing schemes under uncertainty.
While related, their environment is different from this work in several ways. Firstly, pricing

decisions are made in real time, allowing the agent to adapt prices hourly based on system observations. This thesis focuses on day-ahead pricing, where a full 24-hour price schedule is published at the start of the day and remains fixed. This reflects realistic tariff publication demands that charging station operators have. Secondly, the environment of Mahmud et al. [37] does not model spatial competition or network structure between charging stations. In this thesis, EV drivers can choose between geographically proximate charging stations owned by different operators, which captures the competitiveness of the environment. Finally, their work evaluates reinforcement learning methods only, while this thesis compares reinforcement learning agents with transformer-based models within the same simulation environment.
This work is therefore complementary to Mahmud et al. [37] and shifts the focus from real-time control to day-ahead pricing and adds network-level constraints.

4 Data

This section describes the input datasets used in this thesis and the transformations applied to make them suitable for the simulation experiments.

4.1 EV charging session data

The Charging Session Data serves as one of the inputs of the simulation environment. This data will serve as a form of expected demand for a charging station. The data is publicly available from Elaad, which is a Dutch organisation that collects information from public charging stations across the Netherlands. The dataset can be collected from the Elaad charging profile simulator [18]. To get the expected charging station profile for 2025 for a single charge point the following configurations were used:

- Profile type is set to Charge point.
- Number of charge point profiles is set to 1.
- Simulation year is set to 2025.
- Location type is set to public.
- Charging capacity (kW) is set to 11.
- Charging Station capacity (kW) is set to 17.25.
- Smart charging is turned off.
- A different random seed is used for every single generated charge point.

Generating more than one charge point profiles does not seem to work as of October 2025. Therefore, each charging profile needs to be generated individually. After setting the required amount of profiles to 1 and clicking the generate button, a .csv file is generated with the format that can be seen in Table 1.
The data is the predicted demand on a single charge point and is therefore not real-world data. However, it is based on real charging station data and can be used to capture behaviours of EV drivers. The dataset contains the CET time for 2025 per 15-minute interval. Both profile_0_0

datetime	profile_0_0	profile_0_1
2025-01-01T00:00:00+01:00	0.0	0.0
2025-01-01T00:15:00+01:00	0.0	11.0
2025-01-01T00:30:00+01:00	0.0	6.0

Table 1: A sample of the charging session data downloaded from Elaad

and profile_0_1 contain the average amount of kWh charged per 15 minute interval and is transformed further as will be described in the following subsection. To only get one charger profile, the data from profile_0_1 is extracted and used for further processing.

4.1.1 Transforming EV charging data to session data

The raw Elaad profiles are numbered and stored as .csv (`profile_.csv`). Afterwards, these files are then read by a script that tries to find continuous charging sessions. The first timestamp where kWh is given to profile_0_1 is the starting time. The timeslot before the one with 0.0 kWh is given will be set as the ending time of the charging session. The duration in minutes is calculated between the end and start time, along with the total amount of energy delivered and the average kWh. This results in a file that follows the schema that can be seen in Table 2.

source	profile	start	end	duration_min	energy_kwh	avg_kwh
profile_1.csv	profile_1	2025-12-30 18:45:00	2025-12-30 20:00:00	75.0	35.6	7.12
profile_1.csv	profile_1	2025-12-31 12:30:00	2025-12-31 13:45:00	75.0	48.4	9.68
profile_2.csv	profile_2	2025-01-01 09:30:00	2025-01-01 10:15:00	45.0	31.6	10.53
profile_2.csv	profile_2	2025-01-02 08:30:00	2025-01-02 09:00:00	30.0	18.0	9.0

Table 2: A sample from the processed session data

4.1.2 Scaling charging demand based on EV adoption

The Elaad charging profiles generate realistic charging behaviour in 2025. This does not mean they are representative for the period 2018-2024. The experimental setup requires training of dynamic pricing strategies over the period 2018-2024 however. Therefore, it is necessary to scale charging behaviour that reflects historical demand.
To achieve this, statistics on the number of battery electric vehicles (BEV), plug-in hybrid electric vehicles (PHEV) and fuel cell electric vehicles (FCEV) in the Dutch vehicle fleet were taken from the official vehicle fleet dashboard from the Dutch government [41]. January 2025 is taken as the reference point and is defined as representing 100% of total charging demand. For each simulation month between 2018 and 2024, a demand scaling factor is computed as:

$$s_M = \frac{N_{\mathsf{EV}}(M)}{N_{\mathsf{EV}}(\text{Jan 2025})} \tag{4}$$

where $N_{\mathsf{EV}}(M)$ is the total number of registered electric vehicles in the month M.
The expected number of charging sessions in each simulation period is then reduced proportionally by applying this scaling factor. For example, if the number of EVs in March 2023 corresponds to 20% of the January 2025 fleet size, only 20% of the extracted charging sessions are sampled and used in the simulation for that period.
EV registration data is not always available for every month or quarter. In such cases, the most recent available EV count preceding the simulation period is used, which is the last observation.

This results in a realistic growth pattern in charging demand over time, without the need for real or generated data during 2018.

4.2 Wholesale electricity price data

The wholesale electricity prices serve as a baseline for the cost of energy for the charging stations for the simulation environment. The prices are obtained from the European Network of Transmission System Operators for Electricity (ENTSO-E) and correspond to day-ahead market prices for the zone in the Netherlands. For this project, the wholesale prices were collected for the period 2018-2025. The data is available via API and requires a token, which can be freely requested from ENTSO-E. The data is returned in XML format and contains hourly market prices in EUR/MWh. Each record includes the price, time interval and metadata such as unit and currency. Due to API query size limitations, data was retrieved sequentially on a per-month basis for each year. The returned XML files were parsed using the `xml.etree.ElementTree` module, after which all hourly price points were combined into a single dataset.
The raw data returns time in UTC, which needs to be converted to CET to match with the EV charging session data. Furthermore, the raw data may contain some missing hourly values or duplicated hours, which can occur due to API inconsistencies. Firstly, all timestamps were deduplicated. Next, the dataset was merged with an hourly UTC index with all hours from between 2018-2025. Any missing hours were filled forward, with backward-fill being used only at the start of the series in case it was required. Finally, prices were converted from EUR/MWh to EUR/kWh. A sample from the final dataset can be seen in Table 3.

timestamp_utc	timestamp_europe_amsterdam	price_eur_per_mwh	price_eur_per_kwh
2020-12-31 22:00:00+00:00	2020-12-31 23:00:00+01:00	52.26	0.05226
2020-12-31 23:00:00+00:00	2020-12-31 00:00:00+01:00	50.87	0.05087
2021-01-01 00:00:00+00:00	2021-01-01 01:00:00+01:00	48.19	0.04819
2021-01-01 01:00:00+00:00	2021-01-01 02:00:00+01:00	44.68	0.04468

Table 3: A sample of the wholesale data from the ENTSO-E API

5 Methodology

This section provides an overview of the methods used for the final goal of making comparison possible between different dynamic pricing algorithms. The goal is to evaluate how different model types balance revenue generation with grid congestion mitigation, among other constraints. The goal of this study is not to develop a fully optimized pricing strategy, but to demonstrate the application of reinforcement learning in this simulation environment for dynamic EV charging pricing under grid congestion constraints. This work shows a proof-of-concept that investigates the usability of this controlled environment to test different models that have been established in dynamic pricing literature. The focus is therefore on evaluating the overall approach rather than on obtaining an optimal pricing policy or extensive parameter fine-tuning.
The experimental framework is structured around three components:

1. **Simulation environment:** A modified version of the open-source `CharGym` environment is used to simulate the operation of multiple EV charging stations, price-sensitive drivers, and competing stations. This environment provides a controlled but semi-realistic context in which the models can learn or generate pricing decisions.

2. **Model descriptions:** Two model families are compared. The first consists of Reinforcement Learning (RL) agents: Q-learning, Deep Q-Network (DQN), Deep Deterministic Policy Gradient (DDPG), and that learn price adjustments based on trial-and-error interaction with the environment. The second is a transformer-based time-series model that directly predicts daily price profiles using historical and contextual data.

3. **Evaluation metrics:** All models are evaluated using the same multi-objective performance criteria, derived from the custom reward function. These metrics include total revenue, energy delivered during congestion hours, price volatility, and customer retention in the presence of competitors.

5.1 Assumptions

The simulation framework and model comparisons presented in this thesis are based on a set of modelling assumptions. These assumptions are necessary to balance realism and experimental control. They do not fully represent real-world EV charging dynamics, but are a set of rules that capture some relevant mechanisms for dynamic pricing decisions under grid capacity constraints.

5.1.1 Demand and arrivals

EV charging demand is assumed to come from a stochastic stream of independent arrivals. Arrivals during each timestep t are modelled as a Poisson process [28], where the expected arrival rate varies over the day and is derived from synthetic charging session data. This assumes that arrivals are independent across drivers and that high-demand periods show higher variance than low-demand periods. Prices can influence the allocation of demand across charging stations and time, but do not alter the total number of arrivals within a given hour.
Charging demand is further assumed to scale linearly with the size of the charging network. Increasing the number of charging stations increases the number of expected arrivals. This reflects the tendency of high-density charging locations attracting more demand.

5.1.2 User behaviour and price sensitivity

EV drivers are assumed to be partially price-sensitive. If multiple charging stations are accessible within a local area, drivers can choose between providers based on the price differences. This choice behaviour is modelled using logistic response functions and captures diminishing sensitivity to changes [52]. Drivers may delay the start of a charging session by at most one hour if a lower price is announced for the following hour. Long-term strategic behaviour for EV drivers, like learning and anticipation beyond one time step is not modelled.
Price elasticity values reported in the literature are used as a validation reference and not as a calibration target. Empirical elasticity estimates can vary in different locations. Thus, they are assumed to be an appropriate indicator of the magnitude of EV-driver responsiveness to price changes.

5.1.3 Temporal

Pricing decisions are made on a day-ahead basis. At the start of each episode, the charging station operator publishes a full 24-hour price schedule, which is fixed throughout the day. EV

drivers are assumed to have perfect information about these prices. Real-time hourly price updates are not considered in this project.

5.1.4 Charging infrastructure

The charging network is modelled as an undirected graph representing geographical proximity between charging stations. Competition is local. EV drivers can only consider stations within the same connected component of the network. Each charging station can serve at most two EVs at a time, representing an average public charging station in the Netherlands and no waiting queues are modelled.
Charging power per station is assumed to be constant. The state of the battery in the EV is not modelled. A global site capacity constraint represents grid limitations at network level. When aggregated demand exceeds the available energy is proportionally distributed across active sessions. Sessions that depart without receiving their required energy penalize the charging station operator.

5.1.5 Data

Charging demand patterns are derived from synthetic charging profiles generated using the Elaad charging profile simulator [18]. These profiles are not real-world sessions. They are assumed to capture realistic demand patterns. Absolute demand levels are adjusted using historical EV adoption statistics from the Dutch government [41], assuming that public charging demand scales with the number of registered EVs.
Wholesale electricity prices are taken from day-ahead market data and are assumed to represent energy costs. Additional tariffs and price structures are not considered.

5.1.6 Scope

Under these assumptions, the environment intends to be representative of behavioural responses. These results should therefore be interpreted as insights into the relationship between pricing strategies and model choices and do not directly prescribe deployment to the real-world.

5.1.7 Summary of assumptions

For clarity, the assumptions underlying the simulation environment and experiments, as described in Section 5.1, are summarized below:

1. EV charging arrivals are stochastic, independent across drivers, and follow a Poisson process with a time-varying expected arrival rate.

2. Electricity prices can influence charging demand across stations and for the next hour, but do not affect the total number of arrivals within a given hour.

3. Expected charging demand scales linearly with the number of charging stations in the network.

4. EV drivers are partially price-sensitive and probabilistically choose between locally accessible charging stations based on relative price differences.

5. EV drivers may delay the start of a charging session by at most one hour if a lower price is announced for the following hour.
6. Long-term strategic behaviour, learning, and anticipation beyond a single timestep are not modelled.
7. Pricing decisions are made day-ahead and published as a fixed 24-hour price schedule.
8. EV drivers are assumed to have perfect information about announced day-ahead prices.
9. The charging network is represented as an undirected graph capturing local spatial competition.
10. Competition between charging stations is local and EV drivers only consider stations connected via edges.
11. Each charging station can serve at most two EVs at a time and no waiting queues are modelled.
12. Charging power per station is constant and battery state is not modelled.
13. A global site-level capacity constraint represents grid limitations and caps aggregate energy delivery.
14. Charging sessions that depart without receiving their required energy incur a penalty for the charging station operator.
15. Charging demand patterns are derived from synthetic profiles generated by the Elaad charging profile simulator.
16. Public charging demand is assumed to scale proportionally with the number of registered electric vehicles.
17. Wholesale day-ahead electricity prices are assumed to represent marginal energy costs.
18. Network tariffs, balancing costs, and contractual pricing structures are not considered.

5.2 Simulation environment

In this paper, the simulation environment is described as semi-realistic. This means that the model incorporates several key characteristics of real-world EV charging systems, such as stochastic charging demand, day-ahead electricity prices and competing charging operators, while remaining a simplified representation of reality. The EVPriceNet environment is developed to support dynamic pricing strategies and to incorporate semi-realistic physical and economic constraints. The environment provides a controllable testing ground in which different pricing algorithms can be compared under identical conditions. The design is structurally inspired by CharGym [25], but differs in its objective. While CharGym focuses on charging and discharging, this environment uses pricing as the decision variable and models user behaviour as a response to announced prices.

A key component of the environment is that an episode represents one full day, split into 24 time steps per day. At the start of each episode, the agent submits a daily price schedule. This is a vector of 24 price decisions that represent the prices that EV drivers have to pay. The environment then simulates how EV drivers respond to these prices.

5.2.1 Observation space

At the beginning of each episode ($t = 0$), the agent receives a day-ahead summary of the environment and must submit a full 24-hour pricing schedule. During the rest of the episode, the agent does not receive additional state information and cannot adjust its submitted decisions. This design choice reflects day-ahead price publication and allows EV drivers to be aware of pricing for the day. The observations are represented as a vector provided once per episode and contain only information that would be available day-ahead. It consists of four temporal encodings and three sets of 24-hour forecasts:

$$\begin{aligned} o = \big[& \sin\!\big(2\pi \tfrac{\text{dow}}{7}\big)\,,\ \cos\!\big(2\pi \tfrac{\text{dow}}{7}\big)\,, \\ & \sin\!\big(2\pi \tfrac{\text{doy}}{365}\big)\,,\ \cos\!\big(2\pi \tfrac{\text{doy}}{365}\big)\,, \\ & \pi_0^{\text{wh}}, \ldots, \pi_{23}^{\text{wh}}, \\ & \lambda_0, \ldots, \lambda_{23}, \\ & c_0, \ldots, c_{23} \big]. \end{aligned} \tag{5}$$

The first four elements in the observation space provide a cyclic representation of the week and year using sine and cosine transformations. These encodings allow the agent to use structural regularities in charging demand without suffering from discontinuities in the raw time indices. For example, weekdays and weekends follow different usage patterns, and the transition from Sunday ($\text{dow} = 6$) to Monday ($\text{dow} = 0$) should not appear as a large numerical jump. Similarly, seasonal variations such as higher winter charging demand can occur due to lower battery efficiency.

$$\sin\!\big(2\pi \tfrac{\text{dow}}{7}\big)\,,\ \cos\!\big(2\pi \tfrac{\text{dow}}{7}\big)$$

captures the weekly cycle, and

$$\sin\!\big(\tfrac{2\pi\text{doy}}{365}\big)\,,\ \cos\!\big(\tfrac{2\pi\text{doy}}{365}\big)$$

captures the yearly cycle. Both cycles are mapped onto the unit circle. Weekdays that are adjacent in real time are also adjacent in the encoded representation, which improves the learnability of the periodic structure in the environment.
$\pi_0^{\text{wh}}, \ldots, \pi_{23}^{\text{wh}}$ is the hourly wholesale electricity price forecast and is used to compute the cost of supplying the energy. This represents day-ahead pricing, which should be widely available. $\lambda_0, \ldots, \lambda_{23}$ represents the expected arrival rates for the coming 24 hours. The calculations for these will be described in Section 5.2.4. Lastly, there is the congestion indicator that is true between 16:00 and 21:00. These are chosen as peak hours due to a Dutch government campaign to decrease energy consumption in these hours [46]. These are formally defined as:

$$c_t = \begin{cases} 1, & \text{if } t \in [16,\, 21), \\ 0, & \text{otherwise.} \end{cases}$$

It is important to note that the agent does not observe current information about the network of chargers like occupancy, competitor information or graph structure. The operator of charging stations within a network might not have access to this data. Current occupancy and active sessions at timestep $t = 0$ should not have an impact for the pricing strategy for the day. While forecasts of network occupancy and competitor data could improve the decision-making, they are not included in this project. These forecasts are rarely available or reliable and this

project focuses on creating pricing strategies on widely available and realistic data as a charging station operator. This design reflects reality as operators do not have real-time information of every process in the environment. Therefore, the environment can be classified as a partially observable Markov decision process (POMDP) [48].

5.2.2 Action space

After receiving the observations at the beginning of each episode ($t = 0$), the agent submits a vector of 24 prices, one for each hour of the upcoming day:

$$a = (p_0, p_1, \ldots, p_{23}), \qquad p_t \in [0, 1]$$

The agent directly allows a price $p_t \in [0, 1]$. Once the daily price vector is submitted, it remains fixed throughout the day and the agent is no longer allowed to change the prices. This models day-ahead publication and reflects common operational practice. During the simulation, at each hour t, the environment retrieves the corresponding price p_t from the schedule and uses it when calculating rewards.
The upper bound $p_{\max}$ is introduced to prevent unrealistic pricing strategies and reflects real-world constraints on public charging tariffs. In these experiments, a value of 1 EUR/kWh is fixed across all models to ensure comparability. According to ANWB (Algemene Nederlandse Wielrijdersbond) [5], the average prices on a fast charger are between 65 and 90 cents. The upper bound value of 1 EUR/kWh was chosen as a rounded maximum.
Figure 1 summarizes the reinforcement learning loop in the proposed environment. The observation space consists of four components: calendar-based time encodings, the day-ahead wholesale price vector, the expected arrival rates, and the congestion hour indicators. Together, these form the state representation available to the agent at the beginning of each episode. Importantly, since the agent is working in a POMDP the observations do not contain full information of the environment. The agent selects a 24-dimensional action vector corresponding to the day-ahead pricing schedule for the next 24 hours. Once selected, this pricing schedule remains fixed throughout the episode. The environment then transitions hourly. The agent receives a cumulative reward at the end of the episode, further explained in Section 5.2.10. A new observation and action opportunity are provided only at the beginning of the next episode, corresponding to a new day.

5.2.3 Charging network structure

The environment uses an undirected graph G to model proximity and competition. Nodes correspond to a charging station and edges represent geographical proximity. The graph is generated using the file Charging_Network_Graph.py. A fraction of the nodes can be labelled as competitor nodes, while the other nodes will be assigned under ownership of the agent. In the experimental setup used in this paper the ownership of the nodes is split into 50% competitor nodes and 50% agent-owned nodes.
The graph generation requires both a minimum and maximum degree constraint. This is to ensure that a charging station is in close proximity to only a limited number of other charging stations. In the experiments, the minimum degree is set to one and the maximum degree is set to three. This range limits local competition while still allowing alternative charging options within close proximity. This setup will result in a sparse graph with multiple connected components, which can be typically found in real-world charging networks. Figure 2 shows a

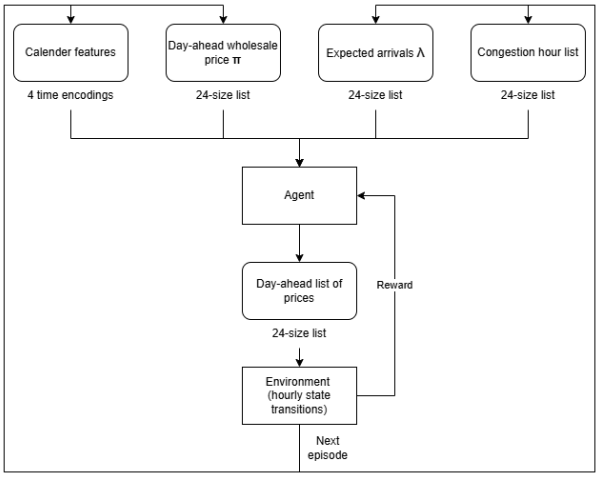

Figure 1: The reinforcement learning feedback loop along with the input and output of the agents

generated graph with 50 nodes, that later is used in the experimental setup. The distribution between own nodes and competitor nodes, minimum and maximum degrees are as described above.

5.2.4 Poisson arrival sampling

To model the stochastic arrival of EV drivers throughout the day, the environment uses a Poisson process [28] that scales with the size of the charging network. The choice for a Poisson distribution is based on two modelling assumptions about charging demand:

- Arrivals are random and independent between drivers.
- Variance grows with expected demand, meaning busy hours naturally exhibit more variability.

From the input data, the average session start rate for that specific day is used as the expected arrival rate λ_t for each day in the simulation. In the experimental setup, λ_t is interpreted as a per-station arrival rate. The total arrival rate is defined as $\lambda_t^{total} = \lambda_t \cdot V$, where V denotes the number of nodes in the charging network. This means that the expected number of arrivals scales with the amount of nodes. Therefore, increasing the number of nodes will increase the number of expected arrivals, and by extension the expected demand.
The environment then samples the actual amount of EVs arriving during hour t using a Poisson distribution. The calculation is as follows:

$$P(N_t = k) = \frac{(\lambda_t^{\text{total}})^k e^{-\lambda_t^{\text{total}}}}{k!} \tag{6}$$

Here $P(N_t = k)$ is the probability that k EVs arrive within hour t. N_t is the number of arrivals during hour t. $k = 0, 1, ...$ and is thus a natural number. λ_t^{total} is the expected number of arrivals during hour t across the entire network. This allows for a higher mean and variability during high demand hours and sparse arrivals during low demand hours. The environment therefore allows fluctuations in EV driver behaviour without using deterministic input profiles, while still being based on actual EV driver behaviour.

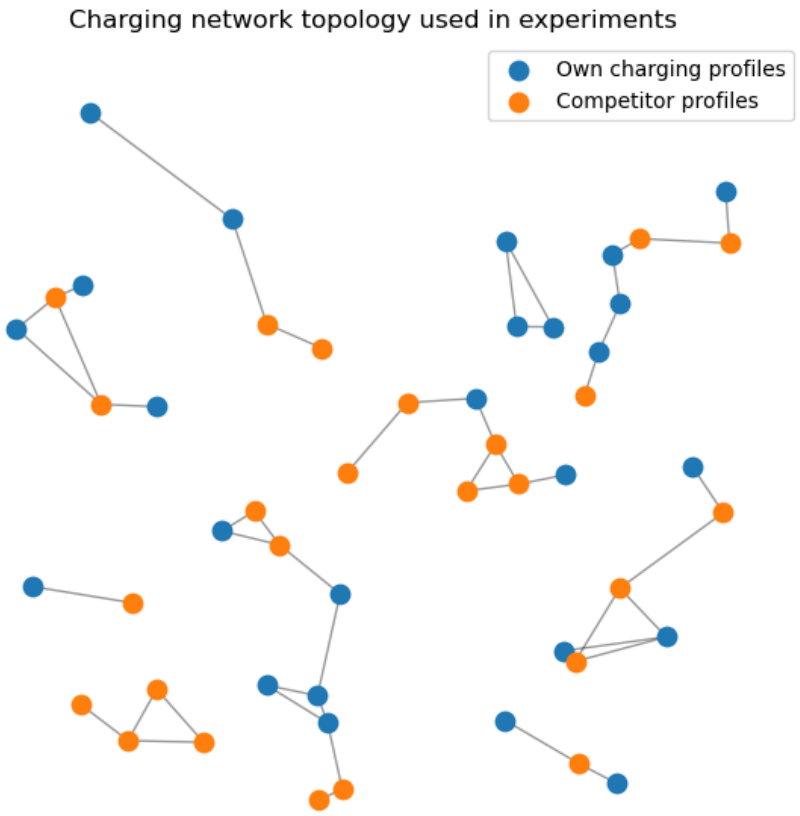

Figure 2: A 50 node graph that represents a charging network, with each node representing a charging station and each edge representing proximity to another charging station, used in the experimental setup and fixed to reduce variability

After calculating N_t, each EV is assigned to one of the connected components of the charging graph. Connected components are selected with a probability that scales with their number of nodes. A larger connected component will therefore attract more demand. This represents a location where companies are more inclined to place more chargers and expect more demand. An example could be close by a busy shopping centre or hospital. Small connected components could represent locations where companies expect lower demand, like sparsely populated rural areas. Afterwards, the EV is assigned to arrive randomly to one of the nodes in the connected component.
To show the variability in charging session profiles, Figure 3 shows two distributions taken from the session data from Elaad [18]. The profiles are for June 25th and December 25th and show large differences in both timing and size of the demand peaks. The Poisson process captures the fact that peak hours have more variability and that off-peak hours should have sparse arrivals.

5.2.5 Pricing-based user choice

Each episode begins with the agent submitting a full daily price schedule $a = (a_1, a_2, ...a_{24})$ where each $a_t \in [0, 1]$ and represents a pricing decision for hour t for all chargers that belong to the agent. This is to model the fact that EV drivers require time ahead to get knowledge of

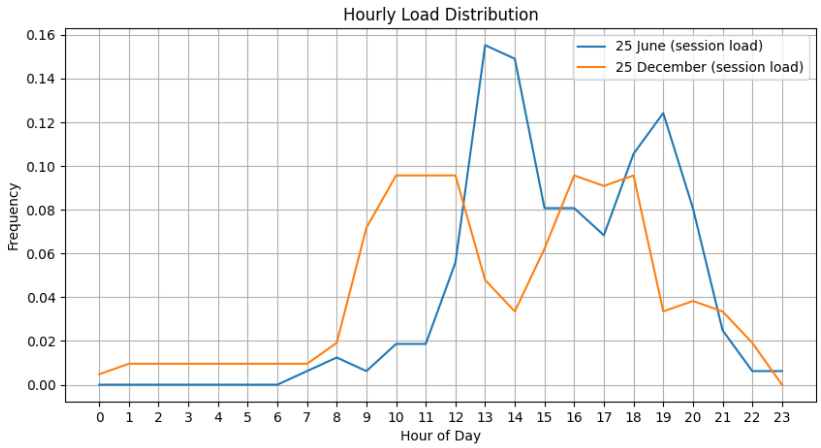

Figure 3: A comparison between the averaged session data of 100 charging station profiles on June 25th and December 25th showcasing differences in timing and magnitude of the demand peaks

the prices for the day.

Each connected component in the charging network represents a local competitive region containing either a mix of competitor-owned and own agent-owned charging stations, only own agent-owned charging stations or only competitor-owned charging stations. If an EV is assigned to a mixed component, it will arrive at a random node within this component. It then checks whether any neighbouring node belongs to a different operator. If all nearby nodes are owned by the same operator as the start node, the EV driver has no alternative charging provider and all reachable prices are identical. The EV driver therefore has to start charging at that location, but they do have the option to delay their session with one hour. In the case that there are chargers nearby of other owners, it needs to decide whether to stay at the node it initially arrived at or move to a nearby node which potentially has cheaper prices. The environment models the probability of choosing the agent's chargers using a binary logistic price-choice model and reflects the diminishing sensitivity to price differences [49]. The probability that an EV selects the agent's station is calculated using this formula:

$$P(\text{choose agent}) = \frac{1}{1 + \exp\big((p_t - p_t^{\text{comp}} - \delta_{\text{comp}})\, k_{\text{comp}}\big)} \tag{7}$$

p_t and p_t^{comp} are the price of timestep t on the chargers owned by the agent or a competitor respectively. A higher p_t relative to p_t^{comp} will result in a smaller chance of an EV driver staying at the agent's chargers. k_{comp} controls that determines how sensitive users are to prices. δ_{comp} is a parameter that allows the tuning of the indifference point between the two choices. For example, if δ_{comp} is set to 0.05 then the EV driver will only consider going to a competitor if there is at least a difference of 0.05. This can occur for example when an EV driver does not consider the price difference high enough to switch to another charging station. After selecting whether to charge at the agent's station, an EV driver will also get the choice to postpone the start of their charging session. EV drivers can therefore choose to shift their charging to a potentially cheaper price in the next hour. The decision to delay is based on the relative price difference between the current hour p_t and the next hour p_{t+1}. If the next hour price is lower, the EV has a probability of postponing the session, which will decrease the costs for the user.

Note that this decision is only available on agent nodes as it is a factor in the reward. There is no calculation for the performance of the competitor nodes and therefore no delay decision is modelled for competitor-owned stations. EV drivers that want to start a session at $t = 23$ do not know the price of the next hour and will therefore never delay. The formula for the delay behaviour is also a logistic function and is written as:

$$P(\text{delay}) = \frac{1}{1 + \exp\big((p_t - p_{t+1} - \delta_{\text{delay}})\, k_{\text{delay}}\big)} \tag{8}$$

The probability for delaying is calculated similarly to the probability of choosing a competitor charging station. The parameters k_{delay} and δ_{delay} are defined independently from k_{comp} and δ_{comp}.

5.2.6 Node occupancy

The environment enforces a rule that each charging node can serve two EVs at a time, which reflects a typical public charger in the Netherlands. If an EV is assigned to a node that is occupied, it will check all neighbouring nodes. It will choose a random node from the list of available ones. This will be the new starting node for the EV and it will then continue with the pricing-based decision process described in Section 5.2.5.
If the arrival node and none of its neighbouring nodes are available, the EV will either try to find another connected component or will move away from the network and will be considered as lost demand. Lost demand reflects EV drivers who would have liked to charge at timestep t but were unable to find an available charger within reasonable proximity. EV drivers that attempt to find another connected component are assumed to be willing to drive further to access a charging station.
Figure 4 illustrates the decision process through which an EV arrives at a charging station and initiates a charging session, as described in Sections 5.2.3-5.2.6. First, it will arrive at a connected component. It will be assigned to a random node within the connected component. If the EV arrives at its destination node, it will find out if it is either occupied or not. In the scenario of an occupied charger, the EV has to look for a new charger and first tries the neighbouring nodes. If those are all also unavailable, it will either try a new connected component or not charge at this time. If the EV driver arrives at a non-occupied charger, it can check if there are multiple charging station providers nearby. If not, it will have to start a session with the only provider in that connected component and stay at its current location. In a mixed connected component, it will have to decide between charging at the agent's chargers or at the competitors. An EV is allowed to delay the start of their session by one hour if the price is lower than at the current hour.

5.2.7 Session representation

When an EV driver commits to charging at an agent-owned node, the environment creates a charging session, which is added to a list of active sessions. A session is represented as a dictionary which contains the following information:
$session = \{remaining\_kwh, dep\_step, node\_id, delayed\_until\}$
Remaining energy is the amount of kWh that the EV driver still requires during this session. The required energy is generated synthetically using a log-normal distribution fitted to historical charging session data obtained from Elaad [18], as described in Section 4.1. The fitted distribution has a mean of $\mu = 20.96$ and a standard deviation of $\sigma = 17.44$. While the

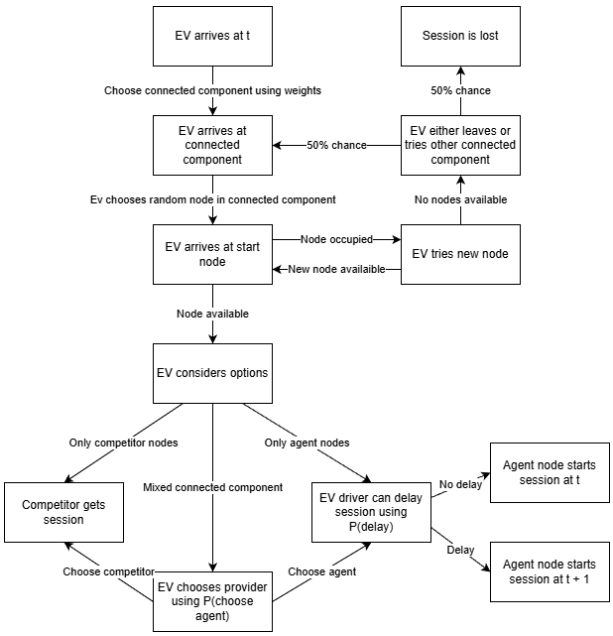

Figure 4: The decision tree that shows how an EV decides on the location and time of the session, either choosing between the agent's or competitor's charging stations and delaying their session from available locations

simulation samples session energies from this distribution, the parameters itself are derived from real-world charging session data. Figure 5 shows the distribution of session energy demand based on the data acquired from Elaad as described in Section 4.1. The distribution is strongly skewed to the right. This indicates a high frequency of short sessions and fewer long ones. Session energy demand is generated based upon this distribution to reflect historical charging data. Session energy values used in the simulation are sampled from the fitted distribution with replacement. To prevent extreme outliers, session energies are clipped using an upper quantile threshold. Values above the 99.5th percentile are capped, while a minimum energy threshold of 4 kWh is enforced to exclude negligible sessions.

Each EV starting a session is assigned a dwell time in hours. The dwell time is sampled from a skewed distribution. Short stays (2-4 hours) are more common and long stays (6-8 hours) occur with decreasing probability. A special case occurs when a session starts after 20:00 to represent overnight charging. The departure time will then always be after 8 hours of charging. dep_step is the departure time, which is formally noted as:

$$\text{dep\_step} = t_{\text{arrival}}^{\text{global}} + \text{dwell\_hours} \tag{9}$$

$t_{arrival}^{global}$ represents the total global timestep that does not get reset every day like the usual t parameter and can therefore simulate charging overnight.

The $node\_id$ variable specifies the specific node within the charging graph where the session will take place. Since a session is only created after the selection process of an EV described in earlier Sections 5.2.3-5.2.5, this node id is fixed and an EV cannot relocate afterwards.

The final part of the session is the $delayed\_until$ parameter. This field specifies the timestep at which the session will take place. The process for calculating a delay is defined in Section 5.2.5. This means that a session can either start in timestep t or timestep $t+1$. If a session has been assigned a delay, energy delivery does not occur until the global timestep reaches $t+1$.

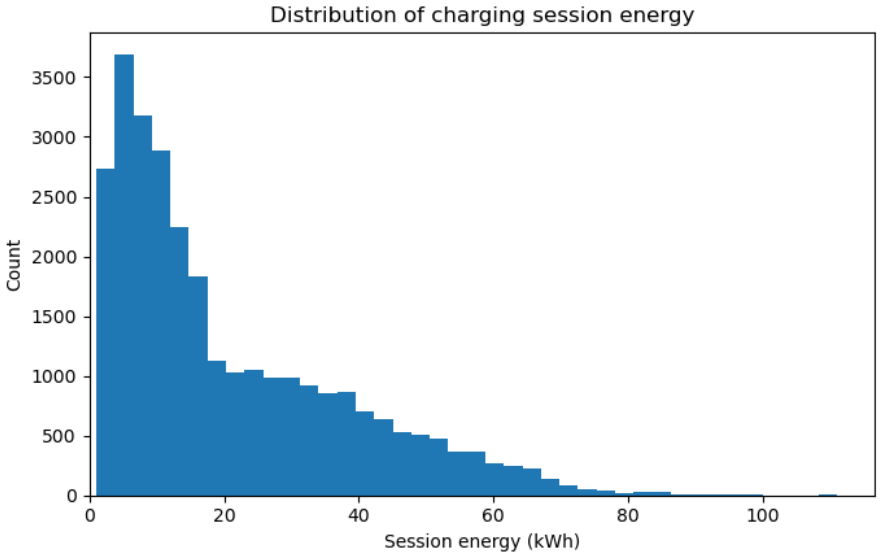

Figure 5: Distribution of session energy demand derived from Elaad [18] data with a right-skewed distribution, indicating that short charging sessions occur frequently while long sessions are less common

5.2.8 Energy delivery and site capacity

When a session is started there are two factors that determine the charging process: the maximum power of a charging station and the site capacity. These two constraints will simulate congestion, which occurs when demand exceeds physical capacity.
Each agent-owned charging node represents a single charging station with a fixed maximum power of 11 kW. This number is the default charging station capacity from the Elaad session data [18] and is therefore used as a baseline in the simulation. The ANWB also cites that the average charging speed for EVs is 7.4-11 kW [4]. This means that at most, the maximum energy that can be delivered to an EV within a single timestep of an hour is 11 kWh. If a session has a remaining energy requirement smaller than this value, only the remaining energy is delivered. A session that is delayed does not receive any energy until the global timestep t^{global} has been reached.
Additionally, the environment has a site capacity which applies to the entire network. The site can deliver at most 11 kWh $* n/2$ kWh in total, with n being the number of nodes. This represents the grid capacity of the network. If the total demand from active sessions exceeds the site capacity, the delivered energy is capped at the site capacity divided by the number of all active sessions. In this scenario, chargers may be capped out due to the grid's maximum throughput, which will cause the chargers to be unable to deliver 11 kWh. This could lead to EV drivers departing without the energy level they would have liked, which penalizes the agent in the form of a shortfall penalty.
To show the effect of the site capacity constraint, Figure 6 gives an example of the relationship between total charging demand and the available site capacity over the course of a day. When the total charging demand remains below the site capacity, each charger can operate at its maximum power and deliver up to 11 kWh per hour. However, when the total demand exceeds the available site capacity, congestion occurs and the available energy must be distributed across all active charging sessions. As a result, individual chargers may receive less energy than

their maximum charging power would allow.

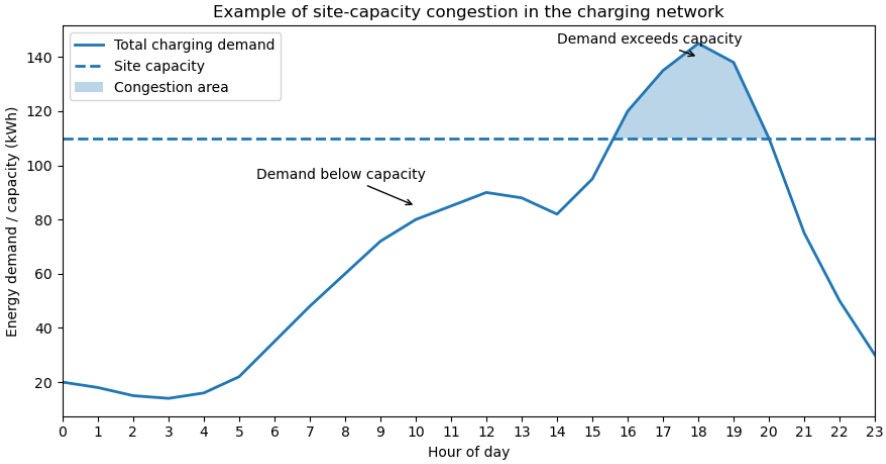

Figure 6: Example of demand exceeding grid capacity, which makes congestion occur and the available energy must then be shared evenly across active charging sessions

5.2.9 Shortfall and session end

At the end of each timestep, the environment checks if an active session has reached its departure time. As mentioned in Section 5.2.7, each session is created with a dwell time that determines the timestep where the EV will leave. If the global clock reaches the departure timestep, the session is considered finished and is removed from the list of active sessions.
If a session departs with remaining unmet energy, this amount will be added as shortfall. Formally, the shortfall for a session is defined as:

$$\text{shortfall}_s = \max\left(0, E_s^{\text{remaining}}\right) \tag{10}$$

This shortfall is only recorded for sessions departing from agent-owned nodes. Competitor-owned sessions do not add to this shortfall, since the agent cannot control incomplete charging there. Shortfall represents EV drivers who were unable to obtain their energy requirement and can either occur due to congestion or due to delaying their session and being unable to get their energy requirement in time. However, even if prices were optimal, the agent may still get shortfall penalties if demand still exceeds grid capacity, which reflects realistic operational constraints in congested charging networks.

5.2.10 Reward function

To compare the performance of the different pricing strategies, all models are evaluated using the same multi-objective performance metrics derived from the simulation environment and the reward formulation. These metrics reflect operational, economic and behavioural parameters that are relevant for charging station operators. Each metric is computed after a whole episode has passed.
The reward function in Equation 11 is defined as a weighted sum of economic, operational and behavioural components. The objective of the reward is not only maximizing revenue, but also to balance profitability with grid congestion mitigation and competitive positioning.

The first term represents the net revenue obtained by the charging station operator. Here p_t is the retail charging price set by the operator at hour t, π_t^{wh} is the wholesale electricity price and $E_t^{\text{delivered}}$ is the total energy delivered by the operator during that hour and stimulates the agent to maximize revenue.
The second term penalizes energy delivery during predefined congestion hours. The indicator function $\mathbf{1}_{t\in\text{congestion hours}}$ equals one if hour t falls within the congestion window between 16:00 and 21:00. Otherwise, it is zero. This penalty discourages charging during peak load periods.
The third term represents the shortfall penalty. $E_t^{\text{shortfall}}$ measures the amount of energy demand that was unmet when a charging session ends. This term encourages pricing strategies that do not overload the system and improves reliable service for EV drivers.
The fourth term penalizes lost charging sessions. S_t^{lost} corresponds to EV drivers who arrive but were unable to start a charging session due to charger occupancy.
The fifth term penalizes price volatility by discouraging large changes between consecutive hourly prices. The absolute difference $|p_t - p_{t-1}|$ reflects customer preferences for predictable and smooth pricing.
The final term penalizes the amount of energy delivered by competitor charging stations, $E_t^{\text{competitor}}$. This term encourages the agent to retain demand within its own network and models the competitiveness of charging networks where multiple charging operators coexist.
The weighting coefficients w_{rev}, w_{cong}, w_{short}, w_{lost}, w_{vol}, w_{comp} control the relative importance of the different terms. These parameters are fixed across all models within the experimental setup to ensure a fair comparison and are selected to reflect a plausible trade-off between revenue, grid impact and quality of service.

$$
\begin{aligned}
R_t = \underbrace{w_{\text{rev}} \cdot (p_t - \pi_t^{\text{wh}})\, E_t^{\text{delivered}}}_{\text{net revenue}} - \underbrace{w_{\text{cong}} \cdot E_t^{\text{delivered}} \cdot \mathbf{1}_{\{t\in\text{congestion hours}\}}}_{\text{congestion penalty}} - \\
\underbrace{w_{\text{short}} \cdot \text{Shortfall}_t}_{\text{shortfall penalty}} - \underbrace{w_{\text{lost}} \cdot S_t^{\text{lost}}}_{\text{lost charging sessions}} - \underbrace{w_{\text{vol}} \cdot |p_t - p_{t-1}|}_{\text{price volatility penalty}} - \underbrace{w_{\text{comp}} \cdot E_t^{\text{competitor}}}_{\text{competitor energy delivered}}
\end{aligned}
\tag{11}
$$

5.3 Model descriptions

This section describes the parameter choices for all approaches already introduced in Sections 2.2-2.3. The goal of this section is not to reintroduce these methods, but to explain the hyperparameter choices, tuning decisions and methodological considerations taken in the implementation of these models.
Across all models, hyperparameters are selected with the goal of demonstrating model performance rather than maximizing performance through extensive tuning. This study aims to assess the suitability of different methods for dynamic EV charging prices.
All models were trained and evaluated using identical simulation settings and reward functions. Performance differences can therefore be attributed to differences in model structure or learning capability. Model-specific parameters are reported in the corresponding subsections.
Furthermore, in this paper the term transformer-based model is used. This refers to models that use the transformer architecture as introduced by Vaswani et al. [50]. This includes learnable methods like the decision transformer and pre-trained foundation models.

5.3.1 Q-learning

Tabular Q-learning serves as a reinforcement learning baseline. As mentioned in Section 5.2.2, the learning task here is a day-ahead pricing problem where the agent commits to a 24-hour price schedule at the start of each episode. The agent is allowed a single action per episode and the consequences are returned throughout the day.
At the beginning of the day, as written in Section 5.2.1, the environment allows the agent to observe encoded calendar data, the day-ahead wholesale electricity price curve for the next day, the expected charging demand based on expected arrivals and the congestion indicators. In this setup, the underlying state and action spaces are high-dimensional and continuous, but are discretized and stored using a hash-based representation for tabular learning.
States and actions are stored using a hash-based representation and the Q-value update seen in Equation 2 is applied using the total daily return as the reward.
The learning rate α is set to 0.25. Exploration is done via an ϵ-greedy strategy with $\epsilon_{\text{start}} = 0.25$ and $\epsilon_{\text{end}} = 0.05$. These values are commonly used configurations for Q-learning and were not extensively tuned. To prevent uncontrolled growth of the Q-table, the number of stored actions per state is limited to 200. When this limit is exceeded, actions with the lowest Q-values are removed.
Due to the time encodings rarely repeating identical combinations of calendar features, tabular Q-learning is not expected to converge to a meaningful policy in this setting. This method is not intended to perform competitively and only serves as a baseline that shows the limitations of tabular Q-learning and highlights the scale and performance of the other methods.

5.3.2 DQN

The input layer of the DQN corresponds to an encoded state vector. This consists of a time-of-day representation as described in Section 5.2.1. There are two time features for the sine and cosine encodings for the day of the week and two more for the sine and cosine encodings for the day of the year. Then there are three 24-hour day-ahead vectors: the wholesale price, expected arrivals and congestion. This results in a 4 + 3 * 24 = 76 dimensional input vector. The agent then selects a full daily price schedule consisting of 24 hourly prices. To enable the use of a DQN, prices are discretized per hour into L evenly spaced levels between 0 and p_{max} with $L = 101$ discrete price levels. This corresponds to discrete price levels spanning the interval $[0.00, 1.00]$. This discretization allows the continuous pricing space to be compatible with the discrete action formulation required by a DQN. Since the pricing action is per hour for each 24 hours, the agent has to select one of the discrete prices. The output of the DQN is therefore 24 * 101 = 2424 Q-values.
Both the online and the target network share the same fully connected architecture. The input layer consists of 76 units corresponding to the state representations described above. This is followed by two hidden layers, the first one with 128 units and the second one 64, both using ReLU activation functions. This network architecture follows the configuration used in prior EV pricing literature [45]. The output layer produces one Q-value for each combination of hour and discrete price level. For a given state, the predicted Q-values are shaped into a (24,101) tensor. The Q-value of a selected daily price schedule is computed as the sum of the selected hourly Q-values, effectively decomposing the daily action-value into independent hourly components. During action selection, the agent selects the price level with the highest Q-value for each hour. Training is performed using experience replay. Each transition consists of the observed state, the selected daily price schedule and the resulting reward. Since each episode corresponds to a

single day and no future state is considered, the discount factor is set to $\gamma = 0$. The network parameters are optimized using the Adam optimizer [27] with a learning rate of 10^{-3}. Huber loss [21] is used to minimize the discrepancy between predicted and target Q-values. Training is performed with mini-batches of 128 samples drawn from a replay buffer with a capacity of 50,000 transitions. Learning starts after an initial warm-up of 365 stored transitions.
To stabilize learning, a target network is maintained and updated every 500 gradient steps. Exploration is handled using an ϵ-greedy strategy applied at the daily decision level, where exploratory actions correspond to selecting random price schedules. The exploration rate starts at 0.25 and decays to 0.05 over the course of the training.

5.3.3 DDPG

The DDPG agent is implemented using the same day-ahead decision structure as the DQN baseline, but operates directly in a continuous action space. The state representation is identical to that used for the DQN and consists of a 76-dimensional input vector composed of temporal encodings and day-ahead vectors.
The actor network maps the encoded state directly to a continuous daily price schedule consisting of 24 hourly prices. The actor outputs normalized actions in the interval $[0, 1]$ by applying a bounded activation function in the output layer, after which the actions are scaled by the maximum allowed price $p_{\max}$ to obtain the final hourly prices. The actor and critic networks both use fully connected architectures with two hidden layers of 128 and 64 units with ReLU activations following prior EV charging dynamic pricing literature [45]. The actor network outputs a 24-dimensional action vector, while the critic network takes as input the concatenation of the state vector and the corresponding action vector. It outputs a scalar Q-value representing the expected total reward for the current day.
Experience replay is used to stabilize training. Each stored transition consists of the observed state, the selected daily price schedule and the resulting reward. Since each episode corresponds to a single day and no future state is considered, the critic is trained to regress the predicted Q-value directly to the observed daily reward, effectively reducing the learning problem to a contextual bandit setting.
The critic network is optimized using mean squared error loss, while the actor is updated by maximizing the critic's estimated Q-value for the actions produced by the current policy. The actor and critic are optimized using the Adam optimizer [27] with learning rates of 10^{-4} and 10^{-3}, respectively. Mini-batches of 128 samples are drawn from the replay buffer with a capacity of 50,000 transitions and learning begins after a warm-up period of 365 transitions.
To improve stability, target networks are maintained for both the actor and the critic. These target networks are updated using soft updates with a smoothing coefficient of $\tau = 0.005$. Exploration is achieved by adding zero-mean Gaussian noise to the normalized action vector during training, encouraging local exploration in the continuous price space. The noisy actions are clipped to the [0,1] interval before scaling, ensuring that all generated prices remain within feasible bounds.

5.3.4 PDDPG

The Prioritized Deep Deterministic Policy Gradient (PDDPG) agent extends the DDPG baseline by incorporating prioritized experience replay to improve sample efficiency and learning stability. The state representation and macro day-ahead decision structure are identical to those used for the DQN and DDPG agents.

The actor and critic networks follow the same architectural design as in the DDPG implementation. To improve learning efficiency, experience replay is implemented using a rank-based prioritized replay buffer. Each stored transition consists of the current state, the selected daily price schedule, the observed daily reward, and a terminal indicator. Since the learning problem reduces to a contextual bandit setting, no bootstrapping over subsequent states is performed. Transitions are sampled according to priorities derived from the magnitude of the temporal-difference error, which in this setting corresponds to the absolute difference between the predicted Q-value and the observed daily reward.
Both actor and critic are optimized using the Adam optimizer [27] with learning rates of 10^{-4} and 10^{-3}, respectively. L2 weight decay is applied to the critic to reduce overfitting. Training is performed using mini-batches of 128 samples drawn from a prioritized replay buffer with a capacity of 50,000 transitions. Learning begins after a warm-up of 365 transitions.
Target networks are updated using soft updates with a smoothing coefficient τ =0.001. Exploration during training is achieved by adding zero-mean Gaussian noise to the normalized action vector, after which actions are clipped to the valid range $[0, 1]$. Gradients for both networks are clipped to a maximum norm of 5.0 to further stabilize training.

5.3.5 Decision transformer

This thesis considers two types of decision transformers. There is the classical decision transformer (DT) [12] and the online decision transformer (ODT) [57]. The difference is how they are trained and deployed. Both the DT and ODT share the same neural architecture based on the transformer architecture introduced by Vaswani et al. [50]. The policy is implemented using a causal transformer architecture following the Decision Transformer framework [12]. The input sequence is projected into an embedding space of dimension 128. This value was chosen to make the model not too computationally expensive. This embedded sequence is then processed by a stack of three transformer encoder layers, each with four attention heads. Finally, a feedforward output head maps the representation at each timestep to a scalar hourly price prediction.
Both models are initially trained offline using a dataset of trajectories generated through interaction within the simulation environment. These trajectories are collected using a chosen policy depending on the experiment. Each trajectory consists of the hourly state sequence, the executed hourly price sequence, and the corresponding realised reward.
From the realised rewards, the return-to-go is computed and used as an explicit conditioning signal during training. The model is trained to minimise the mean squared error between the predicted and executed hourly prices. Optimisation is performed using the Adam optimiser [27] with a learning rate of 10^{-4} and weight decay of 10^{-4}. Mini-batches of 128 trajectories are sampled from a replay buffer. The offline DT is evaluated directly after this pretraining phase and is not updated further.
During offline pretraining, the decision transformer is trained to reproduce the pricing behaviour found in the trajectory dataset using supervised learning. The optimisation objective is the mean squared error between the predicted hourly price and the executed hourly price from the trajectory dataset. To monitor the behaviour of the training process, the evolution of the training loss is tracked throughout the offline pretraining phase. Figure 7 shows the mean training loss across experimental runs, where the shaded region represents the standard deviation across runs. This diagnostic plot is included to show the stability of the offline training procedure used for the decision transformer in the experimental setup.

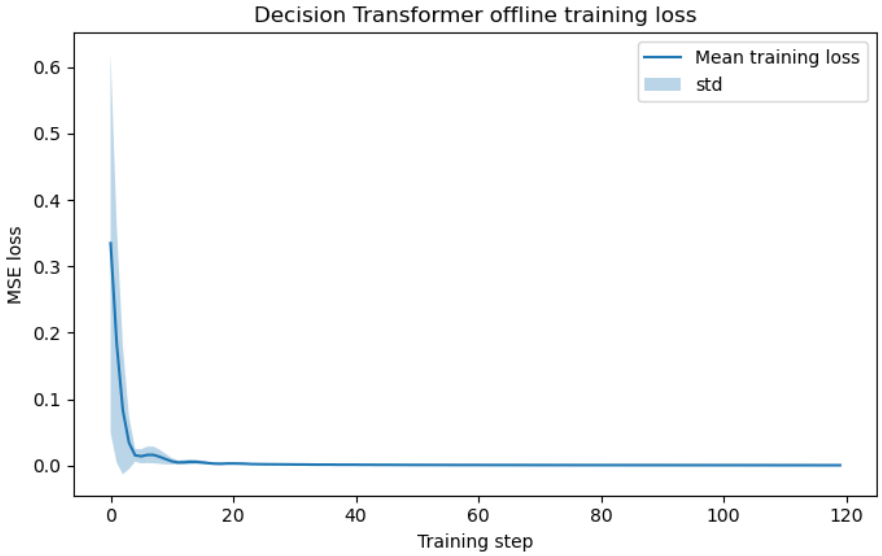

Figure 7: Offline training loss of the decision transformer during supervised pretraining on trajectory data with the line showing the mean loss across runs along with the standard deviation

For the classical decision transformer, this is where its training ends. The offline training phase is the entire training procedure for a classical decision transformer. The resulting model is evaluated without being able to interact with the environment.
The online decision transformer extends the classical DT by allowing additional learning through online interaction. After offline pretraining the ODT is deployed in the environment and generated daily prices at the hourly level like the reinforcement learning methods.
During online interaction, the model generates daily price schedules auto regressively, conditioning on a target return that is set relative to a high percentile of previously observed episode returns. To encourage exploration, the target return is overshot during online data collection. Newly collected trajectories are appended to the replay buffer and the model is periodically fine-tuned on the expanded dataset.
The autoregressive action generation ensures that each hourly price depends on earlier decisions within the same day, allowing the model to learn structured intra-day pricing patterns. Predicted prices are clipped to the feasible interval [0,1]. By combining offline sequence modelling with online fine-tuning, the online decision transformer bridges classical reinforcement learning.
These two variants of decision transformers were chosen to serve as a bridge between the foundation models and reinforcement learning. The classical decision transformer is used in the same experiment as the foundation models, while the online decision transformer is used in the reinforcement learning experiment. They share an architecture and are usable in two different types of experiments, which is why they are chosen in this project.

5.3.6 Time-series Foundation Models

In addition to reinforcement learning and decision transformers, this work evaluates the use of large pre-trained time-series Foundation Models (FM). Unlike RL-based methods, these models are not trained or fine-tuned in the simulation environment. Instead, they rely on historical wholesale electricity prices as their only input to generate a day-ahead price. Two state-of-the-art

time-series foundation models are used in this thesis: Chronos [3] and TimesFM [14]. Both of these models are used in a pure zero-shot setting. No gradients or fine-tuning are performed for these models. They do not observe rewards, charging demand or congestion signals. Chronos uses the *amazon/chronos-t5-base* model and TimesFM the *google/timesfm-1.0-200m* model. Foundation models offer several potential advantages in the context of EV charging price setting. First, they are trained on large and diverse time-series datasets and therefore capture general temporal patterns such as seasonality. A foundation model that has been pre-trained on diverse time-series data may therefore already possess useful representations of such dynamics, even without domain-specific fine-tuning. Second, in a zero-shot context these models can generate forecasts without environment interaction or task-specific training data. Reinforcement learning methods require simulation and exploration to learn a policy, while foundation models can be used without observations from an environment. This could reduce training complexity and computational costs. Finally, foundation models can be very useful in settings where data is limited as they already have been trained during pre-training.
In this work, the foundation models are used as forecasting models. These models are used in a zero-shot manner: given previous historical context, they produce a forecast of future values without interacting with the environment and without receiving feedback. Therefore, they are not trying to optimize a policy and do not change their behaviour based on environment input. The input of the foundation models consists of a fixed-length historical context window containing the most recent H hourly price observations, where H is treated as a methodological parameter. In this work, the context window is chosen to span one week ($H = 168$ hours), which is sufficient to expose the model to both daily and weekly periodicity while keeping the model computationally inexpensive.
The forecasting horizon is fixed to 24 hours, corresponding to the day-ahead decision structure of the pricing environment. The output of the foundation model is therefore a vector of 24 predicted prices, which is interpreted as a pricing action that remains fixed for the day.
The deployment of the foundation models over multiple consecutive days forms a closed-loop process [22]. After each simulated day, the realised retail prices generated by the foundation model are appended to the historical context and used as input for the next day's forecast. The foundation model therefore performs a rolling time-series continuation, where its own previous outputs influence future predictions.

6 Experimental setup

This section describes the experimental setup used to answer the research questions stated in Section 1.1. The experimental setup is designed around the three themes around the three research questions:

- Research question 1: Environment validation
- Research question 2: Training and evaluation of learning-based models
- Research question 3: Evaluation of foundation models

6.1 Environment variables

All experiments are executed in the simulation environment as described in Section 5.2. The parameters used for these experiments can be found in a configuration file called (`config.py`)

to ensure reproducibility. A default set of parameters is used across these experiments and is only changed when specifically mentioned in the following sections to validate the environment. Otherwise, the chosen parameters are as follows:
The years of 2018-2024 are used for model training. No validation year is used due to the high variability in demand patterns and wholesale price dynamics across years. Allocating more years to training provides more representative exposure than a validation split that would reduce the already limited time of the training data. The year 2025 is used as a test set. The experiment is then run 20 times and the results are averaged.
As explained in Section 5.2.3, for this experimental setup, 50% of the network nodes are agent-owned nodes and 50% are competitor nodes. For the experiments in this thesis, 50 nodes are used. This size does not require extensive computational resources and results in 25 agent-owned and 25 competitor-owned nodes. The maximum amount of degrees per node is set to 3 and the minimum is set to 1. As described in Section 5.2.8, the maximum charge speed is set to 11 kW. Two sessions are allowed per node, as this is the average in the Netherlands as described in Section 5.1. To ensure reproducibility, the network is generated using a fixed seed. This fixes node locations, connectivity and competitor assignment across all experiments. The seed chosen for network generation is 42. By holding the network structure constant, the observed performance can be attributed to pricing and decision-making strategies rather than the variability in the charging network. Figure 2 shows the fixed network used in the experiments.
The sessions dataset contains 29,127 sessions. Figure 8 shows that the average number of charging sessions per profile is around 1.31 sessions per day. Given the experiment network of 50 nodes and a multi-year simulation time span of 7 years, there needs to be around $50 * 1.31 * 365 * 7 = 167,352.5$ simulated charging sessions per run. To avoid reused or identical session energy values, the session energy distribution is approximated using a bootstrap buffer of 180,000 samples. The buffer size exceeds the expected number of simulated sessions in a single run, ensuring variability while maintaining computational efficiency.

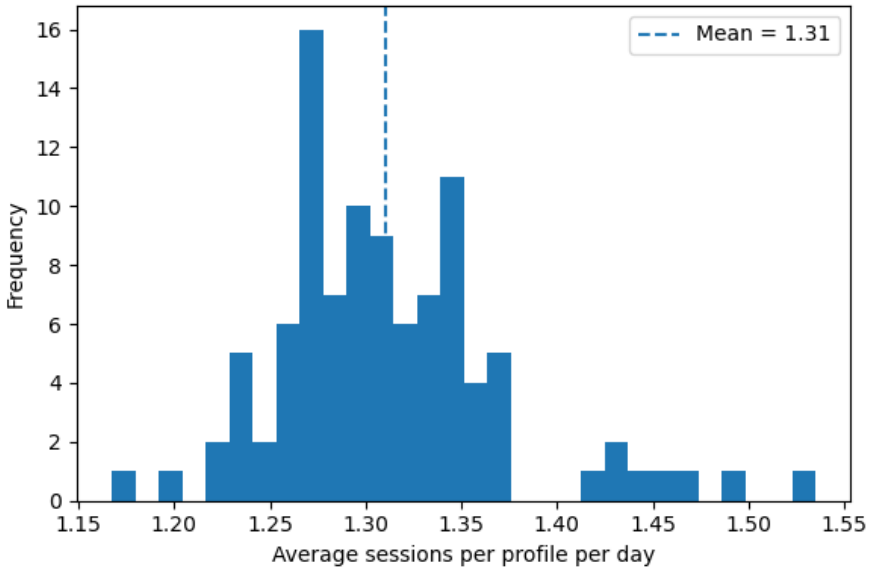

Figure 8: The distribution containing the mean amount of sessions per profile per day

Arrival intensities are derived from the charging session data and normalized by the number

of simulated nodes to ensure scaling demand. A smoothing parameter of $\kappa = 24$ is used to combine specific hourly profiles with the global average profile. This smoothing parameter corresponds to 50/50 weight between the global average profile and one specific day. This prevents every day profile being the same as the global average while also keeping temporal patterns such as seasonality in the demand. The specific formula for smoothing can be found in Equation 12, where λ_t^{day} is the hourly arrival rate for a specific calendar day, $\lambda_t^{\text{global}}$ is the average hourly arrival rate over all days in the dataset and κ is the smoothing parameter controlling the relative weight assigned to the global profile. Finally, a fallback rate of $\lambda = 0.01$ sessions per hour per node is introduced to avoid zero-arrival cases and helps avoid zero-arrival cases that could negatively affect learning stability.

$$\lambda_t = \frac{\lambda_t^{\text{day}} + \left(\frac{\kappa}{24}\right) \lambda_t^{\text{global}}}{1 + \frac{\kappa}{24}}. \tag{12}$$

User charging decisions as described in Section 5.2.5 have two parameters: k_{comp} for relative price sensitivity with competitors, and k_{delay} for price sensitivity for delaying. Both parameters are set to 2.5, based on internal calibration experiments in which retail prices were systematically varied and the resulting price elasticity of demand (PED) was analysed. This value yields a PED that lies within the range reported in empirical studies. The specific values can be found in Table 6.

A fixed competitor price of 45 cents per kWh is used as this is a possible average price per kWh according to the ANWB [5] for public charging stations. The two indifference points are set to ($\delta_{\text{comp}} = 0.05$) and ($\delta_{\text{delay}} = 0.05$) as at least some bias between operators is assumed. This means that EV drivers start to consider other options starting from a price difference of 5 cents.

Finally, the environment reward is defined as a weighted sum of multiple components, which reflect trade-offs faced by a charging operator in practice. These components, as described in Section 5.2.10 are revenue generation, congestion mitigation, shortfall, sessions lost, price volatility and competitor stance. The reward weights chosen represent a balanced operational objective rather than maximizing only one of these components.

Revenue ($w_{\text{rev}} = 0.30$) is assigned the largest weight. This reflects the fundamental economic objective of the operator. However, revenue is not completely dominant, as high revenue maximization can lead to undesirable outcomes such as congestion and volatile pricing.

Congestion penalty ($w_{\text{cong}} = 0.25$) receives a comparably high weight. This choice aligns with the central focus of this study on grid congestion and demand shifting and emphasizes the importance of mitigating peak-hour load.

The shortfall penalty ($w_{\text{short}} = 0.10$), lost charging session penalty ($w_{\text{lost}} = 0.05$) and the price smoothness parameter ($w_{\text{vol}} = 0.10$) are all service-related penalties. Combined, these terms are weighted as heavily as the congestion penalty. This discourages policies that generate revenue at the expense of the quality of service.

Finally, a competitive parameter ($w_{\text{comp}} = 0.20$) is included to prevent strategies that ignore market effects. This component ensures that the agent considers its performance relative to competitors.

6.2 Environment validation experiments

Before comparing algorithms, an assessment must be done whether the environment is stable, reproducible and representative.

- (H_env1) Seed reproducibility: Using the same seed will result in the same results.
- (H_env2) price-volume relationship: Increasing prices should decrease volumes sold and low prices should lead to large volumes sold.
- (H_env3) Congestion-penalty responsiveness: Heavy penalties for congestion should mitigate peak hour consumption.
- (H_env4) The simulated charging demand exhibits an estimated price elasticity of demand (PED) that lies within the empirically reported range for EV charging behaviour.

The experiments to validate these environments are run for 2025 only and are not averaged. The experiments and their results serve as a sample of the dynamics of the environment. All other environment variables are as described in Section 6.1.

6.2.1 Reproducibility

Before interpreting any experimental outcomes, it is essential to verify that the simulation environment is reproducible using fixed seeds. This experiment evaluates hypothesis (H_env1), which states that repeated simulations with the same seed should provide the same results.
Reproducibility is assessed by running the simulation multiple times using the same configuration and one fixed seed for network generation, energy demand and stochastic arrivals. All of these are generated using a fixed seed that is reused across all runs.
Hypothesis (H_env1) is supported if all reported metrics are identical across runs under the same seed configuration. This validation step ensures that observed differences in later experiments can be attributed to pricing strategies rather than random variation in the simulation environment.

6.2.2 Price-volume relationship

To validate whether the simulated environment exhibits a realistic price-volume relationship, a static price experiment is conducted. In this experiment, the agent applies a constant retail price for all hours and days.
The experiment evaluates a discrete set of prices ranging from 0.00 to 1.00 EUR/kWh in increments of 20 cents. The environment is simulated for only the test year of 2025 using the fixed charging network for every price level. All other environment variables are kept as described in Section 6.1.
The expected behaviour under hypothesis (H_env2) is a decrease in charging demand as prices increase. At very low prices, charging demand should increase significantly and EV drivers are expected to be more willing to switch towards the agent-owned chargers than competitor stations. At higher prices, demand should decrease as users would choose competitors. The results can be found in Table 4.

6.2.3 Congestion-penalty response

This experiment evaluates whether the environment responds appropriately to incentives aimed at reducing peak-hour charging demand. Hypothesis (H_env3) is tested here, which states that stronger congestion-related penalties should lead to reduced energy consumption during peak hours. The congestion window is fixed to 16:00-21:00 [46]. The same static price is used

as the competitor of 45 cents per kWh. A fixed surcharge is placed in this interval only for the prices of the agent, ranging from 0.00 to 55 cents per kWh. For each surcharge level, the environment is simulated for the full test year of 2025.
The primary evaluation metric for this experiment is the total energy delivered during congestion hours. Hypothesis (H_env3) is supported if increasing peak surcharges lead to a reduction in peak energy consumption at agent-owned stations. This would indicate that users respond to the surcharges and would rather delay their charging behaviour. Table 5 shows the results of this experiment.

6.2.4 PED

The final environment validation experiment tests whether the simulated charging demand exhibits a realistic price elasticity of demand (PED), as stated in hypothesis (H_env4). Rather than focusing on absolute demand levels, this experiment evaluates whether the responsiveness of demand to price changes falls within empirically reported ranges for EV charging behaviour [29] [32].
The PED analysis is based on the static price experiment described earlier in Section 6.2.2. For every set of prices, the total number of charging sessions at agent-owned stations is computed using the elasticity formulation between price points in Equation 1.
The resulting PED values are computed over multiple price intervals. These values are reported in Table 6. Hypothesis (H_env4) is considered supported if the estimated PED values fall within a believable range reported in literature. This experiment does not aim to match a specific value, but rather to verify that the simulated demand response is of a realistic order of magnitude, ensuring that pricing incentives are plausible behavioural responses.

6.3 Reinforcement learning comparison

This subsection describes the experimental setup used to compare the different learning-based strategies. The goal of this comparison is to assess the performance of the algorithms under the same environmental conditions as described earlier.
This experimental setup was designed from the perspective of a charge point operator without prior reinforcement learning. The operator is assumed to have access to historical charging sessions, day-ahead wholesale prices and some form of demand forecast. The operator is not assumed to have detailed insights into competitor strategies. As a result, all learning-based methods are evaluated under the same informational constraints, reflecting the information a charge point operator should have available.
Four experiments are conducted for the learning-based strategies. All of them correspond to different scenarios. First, all environment variables are set as described in Section 6.1. This scenario corresponds to a plausible weight distribution for the different reward components from Section 5.2.10. Second, only the revenue weight is set to 1, while all other weights are set to 0. This represents a scenario where the agent is only trying to maximize revenue. Third, the environment variables are set again as described in Section 6.1. The difference this time is that the competitor is able to use reinforcement learning themselves and is only interested in maximizing their revenue. The competitor will use the same DQN architecture as described in Section 5.3.2. This model was chosen due to its simplicity and serves as a baseline model. This environment now becomes a competitive setting between two self-interested agents that try to optimize their own objective, which can be defined as a non-cooperative Markov game

[34]. Each agent's strategy is dependent on the behaviour of the other agent and emergent behaviour arises through their interactions. Fourth, the agent has its revenue weight set to 1 and all other weights set to 0, meaning that both agents are only interested in maximizing revenue.
The actions taken by the competitor are the same as the action rules described in Section 5.2.2 and will remain invisible by the agent. The competitor is also not able to see the actions taken by the agent, which means they will have to price their strategy around this. This is done as in a real-world context, a charging station operator will not always have perfect information on the competitor's pricing strategy.
The charging network topology and Poisson sampled sessions is kept consistent throughout all of these experiments. A list of seeds is used to generate all other stochastic behaviour like demand. The number of seeds in this list is determined by the number of runs. For each one of the 20 independent runs in this experiment, a different seed is used. No validation set is used, as increasing the training duration seems to impact the performance significantly. Generalization performance is evaluated exclusively on the test year 2025.
All reinforcement learning algorithms are trained using their respective hyperparameters described in Section 5.3, unless stated otherwise. Hyperparameters are kept fixed across runs. The models participating in this experiment are Q-learning, DQN, DDPG, PDDPG and ODT.
For evaluation, the training curves of the agents are recorded, along with their performance over the full test year. A baseline pricing strategy, namely a static price of 45 cents, is also included in the same environment configuration and its performance is recorded in both the training curves and in the test year. This experimental setup aims for a transparent comparison of the learning-based pricing strategies.
Statistical significance between pricing strategies is assessed using a Friedman test, followed by a Nemenyi post-hoc test for pairwise comparisons [15]. For each pricing strategy, the performance on the test year 2025 over 20 independent runs with identical random seeds is used for this. The resulting test mean rewards are treated as paired observations across models. This approach allows the comparison of multiple models evaluated on the same test setup and does not assume normality of the reward distributions. These results are shown in Appendix A in the form of critical difference plots.
This experimental setup is used to reject or accept H_RL: *Policy-gradient-based reinforcement learning methods (DDPG and PDDPG) achieve higher test performance than value-based methods (Q-learning and DQN) in the dynamic EV charging pricing task.* This result is expected because the agent operates in a high-dimensional continuous action space. Actor-critic methods can directly optimize in continuous action spaces and this result would be in line with Qiu et al. [45].

6.4 Transformer Models comparison

This experimental setup is used to evaluate the transformed-based models. No evaluation is performed on the learning dynamics here, like in the reinforcement learning comparison. The objective here is to examine how the foundation models can be applied in a zero-shot manner within the dynamic pricing environment.
The foundation models are evaluated under the same environmental conditions as in Section 6.1. The same charging network topology, demand process and seeds are used as in the reinforcement learning experiments. This ensures that the performance differences can be attributed to the model differences rather than changes in the environment.

The foundation model experiments focus on a univariate price continuation. A fixed historical price trajectory is generated by the DDPG or PDDPG agent per seed, dependent on the method that achieved a higher mean on the test year. All actions of these methods are recorded for the training years of 2018-2024. For the test year 2025, the foundation models generate daily price schedules by recursively extending this historical price sequence. At the beginning of each day, the model receives a fixed length context window containing the most recent 168 hourly prices (one week) and produces a 24-hour price trajectory, which is then applied to the environment. The generated prices are appended to the historical price sequence and used as input for the next days, resulting in a closed-loop rollout over the entire test year.
Two time-series foundation models are considered in this setting: Chronos [3] and TimesFM [31]. Both models are used without any fine-tuning and operate purely in a zero-shot manner. Evaluation is based on the aggregated performance over 2025 and averaged over 20 runs.
Next to the foundation models, a classical decision transformer is included. The decision transformer is trained on the same historical price trajectories generated by the DDPG or PDDPG agent during 2018-2024 and learns to reproduce the observed pricing behaviour. The decision transformer is deployed in the same closed-loop continuation setting as the foundation models. This allows the decision transformer architecture to be used in both the learning experiments and the transformer experiments. This means that the classical decision transformer is not used in a zero-shot manner and can be classified as supervised learning. The decision transformer is used as a baseline from the same model family to compare with the foundation models.
Two configurations from the reinforcement learning experiments are used. Only the static competitor variants are applied in this experimental setup. The models are evaluated using weights described in Section 6.1 and in pure revenue. Important to note is that these weights do not train or fine-tune the models in this experiment and are only used to evaluate the generated price schedules.
Performance is reported over the test year. The flat tariff and the DDPG or PDDPG results for the test year are also included to serve as a baseline. This allows for a comparison between the zero-shot foundation models, the supervised learning decision transformer and the reinforcement learning approach over the test year.
This experimental setup is used to accept or reject H_FM: *Zero-shot foundation models, evaluated via closed-loop continuation on the 2025 test year, achieve higher performance than the best-performing reinforcement learning policy trained on historical data.* This hypothesis is motivated by the ability of large pre-trained time-series models to adapt to the structural patterns in electricity price dynamics. It is expected that the foundation models are able to generalize on this task and perform better than policies optimized through environment-specific rewards.

7 Results

This section presents the results of the experiments described in Section 6. The results are organized according to the research questions and their associated hypotheses. First, the environment validation experiments evaluate reproducibility, price responsiveness, congestion sensitivity, and demand elasticity. Next, the performance of reinforcement learning agents is reported and compared against baseline pricing strategies and transformer-based models. Finally, the performance of reinforcement learning agents is compared with foundation models in the

simulation test year.

7.1 Environment validation results

All of the experiments in this subsection address Research Question 1 by evaluating whether the simulation environment is stable, reproducible and representative. Before comparing learning-based pricing strategies, it is essential to establish that the environment responds consistently to pricing signals and congestion incentives and that the results are realistic.
A series of environment validation experiments is therefore conducted. These experiments are run for only the simulation year and ensure that the following performance comparisons reflect the differences in the decision-making strategies rather than inconsistencies in the environment.

7.1.1 Reproducibility (H_env1)

To assess the reproducibility of the simulation environment, identical experiment configurations are run across multiple random seeds for an agent with a flat tariff of 45 cents. The resulting total reward, revenue, energy delivered and number of charging sessions remains the same across runs using the same seed. The environment is therefore consistent and can be used to create reproducible outcomes under identical policies. This allows for a fair comparison between different methods used for generating pricing strategies. Overall, these findings support H_env1.

7.1.2 Price–volume relationship (H_env2)

Table 4 presents the aggregated results of the static price experiment. For each price level, the table reports the total margin, delivered energy, and number of charging sessions for both the agent and the competitor. As the agent's price increases from 0.00 to 1.00 euros, a clear decline can be seen in both the total energy delivered and the number of charging sessions at agent-owned stations. Delivered energy decreases from 296 MWh at a zero price to 161 MWh at 1.00 EUR/kWh. The number of sessions drop from 19,409 to 9,815 over the same range. Competitor stations see an increase in both energy delivery and charging sessions, which is due to the EV drivers opting for the cheaper option. Despite the decline, the agent's total margin increases with the higher prices.
These results confirm that the environment shows a consistent relationship between price and demand. Extremely low prices result in high charging volumes but poor profitability, while higher prices shifts demand towards competitors. The observed trend is consistent with the inelastic nature of electricity demand [29, 32].

Static price	0.00	0.20	0.40	0.60	0.80	1.00
Margin (EUR)	-27,209	29,095	72,876	105,209	135,300	146,266
MWh delivered (agent)	296	270	236	207	191	161
MWh delivered (competitor)	192	219	257	288	304	336
Sessions (agent)	19,409	17,750	15,623	12,961	11,774	9,815
Sessions (competitor)	10,416	11,926	14,301	16,153	17,290	19,045

Table 4: Margin table for the agent depending on its static price used compared to the static competitor price of 45 cents, along with the amount of MWh delivered and sessions for both the agent and its competitor

7.1.3 Congestion-penalty response (H_env3)

Table 5 reports the results of the peak surcharge experiment. The agent here applies an additional price surcharge during congestion hours on top of a static baseline price of 45 cents per kWh. For each surcharge level, the total amount of energy delivered during peak hours is shown, disaggregated between agent-owned and competitor-owned charging stations.
As the applied peak surcharge increases, peak-hour energy delivered at agent-owned stations decreases. It decreases from 84 MWh at zero surcharge to 69 MWh at a surcharge of 55 cents per kWh. Over the same range, peak-hour energy delivered at competitor stations increases. However, the total peak-hour energy delivered across the entire network actually increases in this experiment. It varies between 193 MWh and 204 MWh. This indicates that the applied surcharge primarily affects the allocation of charging demand between operators rather than the total amount of peak-hour charging demand. These results demonstrate that implementing surcharges on flat tariffs does not reduce peak-hour consumption over the network and only shifts demand towards competitors.

Peak Surcharge	0.00	0.05	0.10	0.20	0.40	0.55
Peak MWh delivered	193	193	195	195	199	204
Peak MWh delivered agent	84	80	80	78	73	69
Peak MWh delivered competitor	109	113	115	117	126	135

Table 5: Peak total MWh delivered for different surcharges used during peak hours for the agent and its competitor using a static price of 45 cents

7.1.4 Price elasticity of demand (H_env4)

Table 6 reports the estimated price elasticity of demand (PED) for charging sessions, computed using arc elasticity from Equation 1 between price intervals of the static price experiment.
The estimated PED values vary across the price intervals. At the lowest price levels of 0.00-0.20 €/kWh, the estimated elasticity is close to zero. As prices increase, the magnitude of the elasticity becomes larger. In the mid-range price intervals, PED values lie between -0.191 and -0.466. At the highest price interval, the estimated PED reaches -0.817, which indicates a strong demand response.

Price interval	PED (sessions)
$0.00 \rightarrow 0.20$	-0.045
$0.20 \rightarrow 0.40$	-0.191
$0.40 \rightarrow 0.60$	-0.466
$0.60 \rightarrow 0.80$	-0.336
$0.80 \rightarrow 1.00$	-0.817

Table 6: The estimated PED values based on the price levels in Table 4

Taken together, the results of the environment validation experiments indicate that the simulation environment is stable and reproducible, responds consistently to pricing signals, and exhibits realistic demand elasticity. While static peak surcharges do not reduce aggregate peak-hour demand at the network level, this limitation is structural rather than stochastic.

7.2 Learning models results

This subsection presents the results of the learning-based pricing methods. These results include all reinforcement learning algorithms and the online decision transformer. The averaged training curves can be seen for 2018-2024, along with the final averaged reward of the test year 2025.

7.2.1 Multi-objective against static competitor

Figure 9 shows the evolution of the average episodic reward for six pricing strategies: DDPG, DQN, PDDPG, Q-learning, flat tariff baseline and the online variant of the decision transformer. Solid lines show the mean reward across runs and shaded regions represent variability.
All learning-based methods, as well as the flat tariff baseline, show similar learning trajectories. Their reward curves often overlap and follow a downward trend. Rewards are negative here due to all the penalty terms such as the congestion penalty, shortfall penalty and service quality terms. Differences between these methods during training remain small relative to their overall reward and their variability is limited. The environment affects the total reward of these methods greatly. Around 1800 training steps, the DDPG begins to diverge from the other learning methods and continuously outperforms them in training. Because the figure shows standard deviation across runs and is smoothed over a rolling window for readability, the shaded regions appear visually similar across methods. This suggests that the variance and reward between methods are of a comparable order of magnitude.
Table 7 reports the aggregated performance of all models on test year 2025, averaged over the 20 independent runs. The results are expressed as the mean and standard deviation of the reward per episode after the test year. The test results show that the DDPG, DQN, PDDPG, Q-learning and flat tariff baseline achieve rewards that are relatively close. The highest mean value comes from the DDPG with -201.01 and a standard deviation of 5.69. The ODT shows the smallest variance, while the PDDPG shows the highest. The ODT also shows the lowest average test mean and performs worse in test mean than the flat tariff baseline, which performs a little better in mean reward. Notably, the standard deviation between methods is often larger than the gap between methods. This happens with the three worst performing methods of the PDDPG, ODT and flat tariff baseline. The same phenomenon occurs with Q-learning and the DQN. Interestingly, unlike the results of Qiu et al. [45], which used a revenue only setup, the DDPG outperforms the PDDPG in the test year. The PDDPG performs similarly to the DQN and even has the highest standard deviation compared to all these methods.
The reward breakdown in Figure 10 shows the average performance of the DDPG averaged over 20 runs for the different reward components. With reward_rev being the revenue the agent has earned, cong_pen being the congestion penalty, short_pen being the shortfall penalty, lost_pen being the lost sessions penalty, vol_pen being the volatility penalty and comp_pen being the penalty for competitor charging. In this figure, it can be seen that as the amount of episodes increases and the amount of demand scales with the years, there is more revenue earned, as well as more penalties for shortfall, congestion and competitor charging. The two components that do not grow that much with the amount of episodes are the volatility and lost sessions penalty. However, their variance does seem to increase as the amount of episodes go on. The scale of the lost session penalty is very small and is most of the training time exactly 0. This small scale implies that lost sessions are a rare occurrence when taken over 20 runs and might occur only in the later episodes. The volatility penalty fluctuates, which indicates that the agent will try to lower this penalty, but could also focus its actions towards lowering one of the other penalties.

Overall, these results indicate that a static competitor and a multi-objective reward show only small differences in performance between the methods. Still, the DDPG performs slightly better than other methods. However, Figure 15 shows that the DDPG does not perform statistically significantly better than the other methods used in this setup and is equivalent to the DQN.

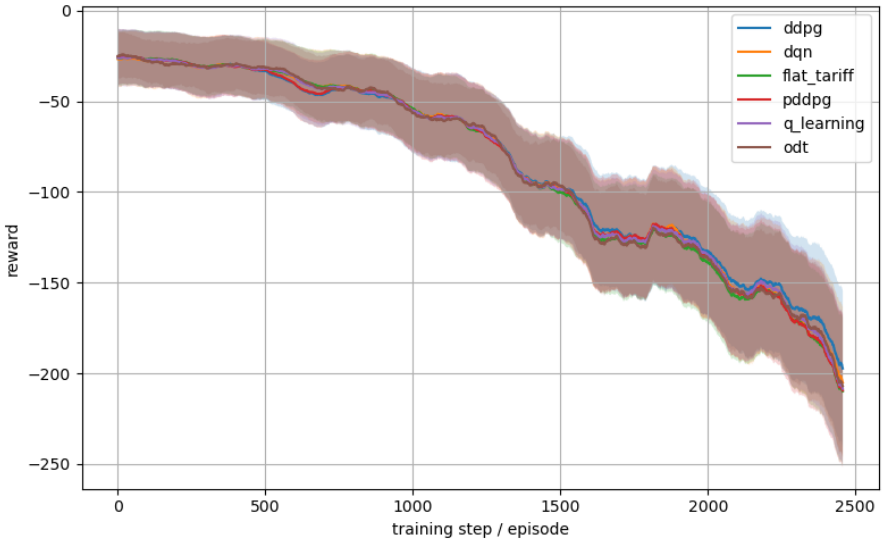

Figure 9: Training curve comparison of different learning-based methods for 2018-2024, using the weights described in Section 5.2.10 against a competitor using a flat tariff of 45 cents, averaged over 20 runs

Model	Test mean	Test std
DDPG	-201.01	5.69
DQN	-209.32	3.88
Flat tariff	-217.17	2.18
ODT	-217.56	1.69
PDDPG	-215.85	8.82
Q-learning	-211.12	2.04

Table 7: The result comparison of different learning-based models on the test year 2025, using the weights described in Section 5.2.10 against a competitor using a flat tariff of 45 cents, averaged over 20 runs

7.2.2 Multi-objective against RL competitor

This experiment evaluates the pricing strategies under a multi-objective reward function when the competitor is no longer static, but is also able to adapt its prices using a DQN for revenue maximization. This setting introduces a strategic opponent, increasing the difficulty of the task and reflects realistic competitive behaviour. This situation can be defined as a non-cooperative Markov game [34].
Figure 11 shows the training curves averaged over 20 runs. These training curves are very similar to the curves in Figure 9. The biggest difference that can be seen is that the gap between the DDPG and other learning methods is slightly larger than in the multi-objective

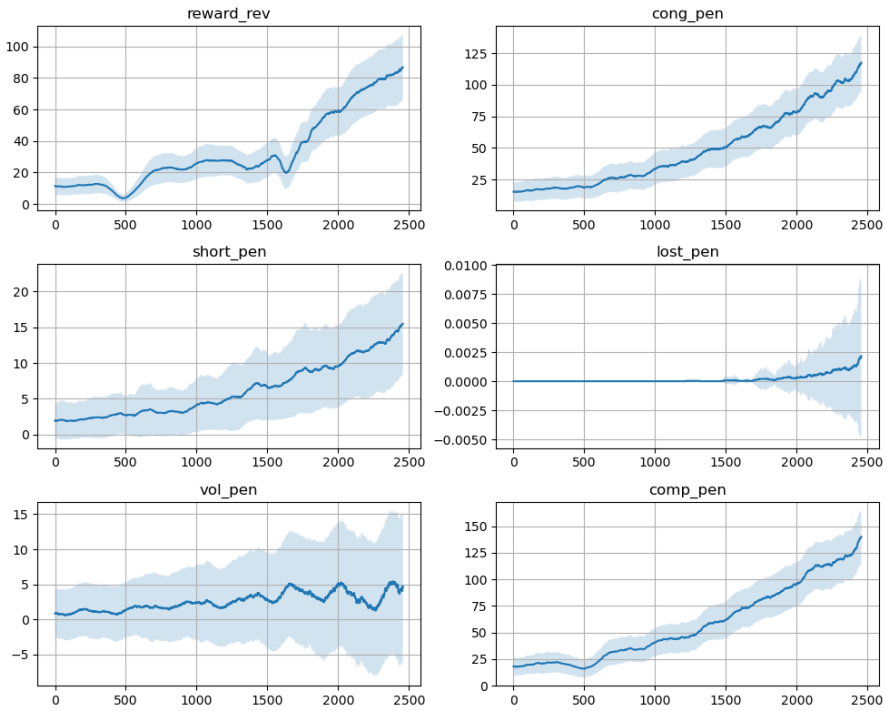

Figure 10: Breakdown of reward over training steps for the DDPG against a static competitor averaged over 20 runs

setup. The shaded regions still largely overlap, but are slightly wider when compared to the previous setup. The performance of the models will therefore likely still be very similar in this setup, with possibly a little more variance.

Table 8 reports the test performance of this setup. The DDPG achieves the highest mean test reward (-179.52), followed by PDDPG (-186.67) and DQN (-188.47). Compared to the previous results, the PDDPG overtook both q-learning and the DQN. The ODT now performs better than the flat tariff baseline. Notably, all methods achieve a higher test mean in this setup than in the static-competitor setup. This does not indicate that these methods perform better in this harder setup. An example for this phenomenon could be that the competitor, now able to adjust its prices, gets more sessions during congestion hours. This could allow the agent to avoid getting penalized for the congestion penalty, since the agent is unable to see the energy sold by competitors. The variance is overall higher in this setup. The standard deviation here is for all methods higher than its difference with the worst performing method of the flat tariff baseline.

Figure 12 shows the performance of the DDPG in this setup for the different reward components. As can be seen when comparing to Figure 10, the breakdown is very similar. The line graphs are almost always identical, except for the volatility penalty, which now shows a clear downwards trend on a much larger scale. The congestion penalty also operates on a much larger scale now. This is likely the result of the competitor setting higher prices, which makes EV drivers charge at the agent's nodes. The shaded area in the congestion penalty plot is also wider than in the previous setup. Furthermore, the reward revenue ends slightly higher on average in this setup than before. This breakdown implies that the main contribution to the overall higher reward in this setup is likely due to the increase in revenue as compared to the previous setup. Compared to the previous experiment, Figure 16 shows that the DDPG and PDDPG, DQN,

and Q-learning are statistically equivalent in this setup. In this harder setup, the PDDPG performs comparatively better than in the static competitor scenario, which allows it now to place second in test mean.

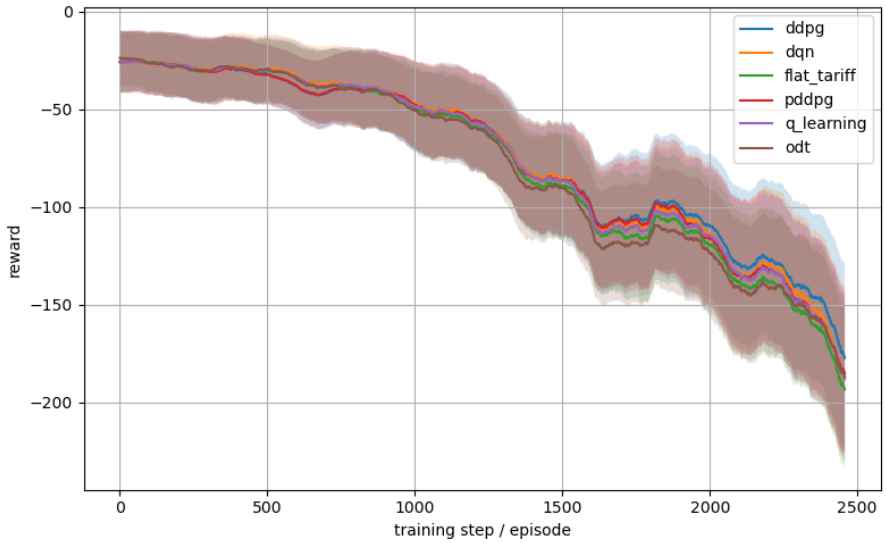

Figure 11: Training curve comparison of different learning-based methods for 2018-2024, using the weights described in Section 5.2.10 against a competitor using a DQN to maximize their revenue, averaged over 20 runs

Model	Test mean	Test std
DDPG	-179.52	23.30
DQN	-188.47	18.37
Flat tariff	-197.24	16.98
ODT	-194.92	4.98
PDDPG	-186.67	18.23
Q-learning	-189.58	16.32

Table 8: The result comparison of different learning-based models on the test year 2025, using the weights described in Section 5.2.10 against a competitor using a DQN to optimize their revenue

7.2.3 Revenue only against static competitor

This section shows the results for the setup where only revenue is optimized. All other components in the reward function described in Section 5.2.10 have been set to 0 and only $w_{\text{rev}} = 1$.
Figure 13 shows that the learning methods have an upward learning trajectory in this setup. After an initial exploration phase, both the DDPG and PDDPG consistently outperform the other approaches in training. Their learning curves separate before the 500 episode mark and continue to diverge. This could indicate that these two actor-critic methods are more effective at exploiting this environment when revenue is the only objective. Compared to the multi-objective setting against a static competitor, the variance in this setting seems to be overall higher,

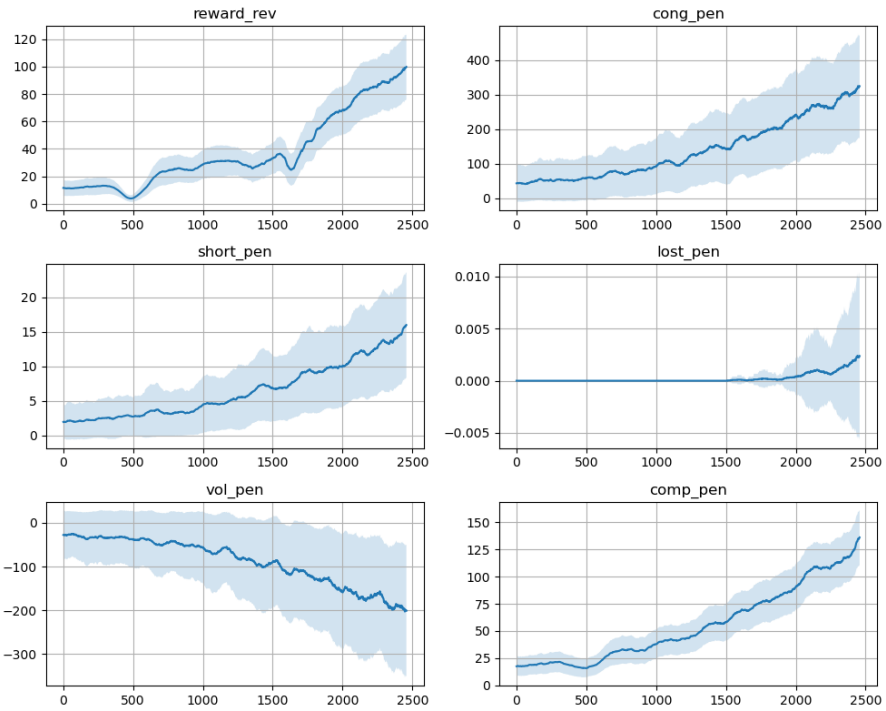

Figure 12: Breakdown of reward over training steps for the DDPG against a competitor using a DQN to maximize revenue averaged over 20 runs

as is represented by the wider range of different shaded areas. It is now more visible which areas belong to which method, which indicates that there is now a more clear difference in performance than in the previous setups. The DQN and Q-learning also show improvement over time, but converge to lower final reward levels. As expected of a high-dimensional and continuous state-action space, these methods perform worse than the state-of-the-art methods. Finally, the online decision transformer and the flat tariff baseline method converge around the same level.

Table 9 shows the aggregated performance for the pricing strategies of 2025 where the reward function only takes revenue into consideration. The PDDPG and DDPG models achieve the highest average test mean, with 389.46 and 376.40 respectively. Both methods also show the highest standard deviation: 20.09 and 37.31. The standard deviations of these two methods are larger than the gap between their performance. The DQN follows up with a lower test mean of 321.87, with relatively low standard deviation of 11.11. Q-learning performs worse than the DQN and shows a very low comparative standard deviation of 3.07. The flat tariff baseline achieves a mean test reward of 248.53 and shows very little variability, which is expected of a static price method. The online decision transformer outperforms the flat tariff baseline here and shows a standard deviation similar to Q-learning. This setup corresponds to Qiu et al. [45] and the performance of the methods they used also aligns with the results in this experiment. They showed that in their case, the best performing methods were in order: PDDPG, DDPG, DQN, and Q-learning. This ranking also holds true when looking at the mean performance in the test year. However, Figure 17 shows that the PDDPG does not show statistical significance compared to the DDPG in this setup.

Note that the test mean shown in Table 9 is revenue only. This means that these values can directly correspond to monetary value. However, these values are the mean reward per day.

An overview of the total revenue in the test year, along with other metrics, can be found in Section 7.2.5 for the PDDPG.

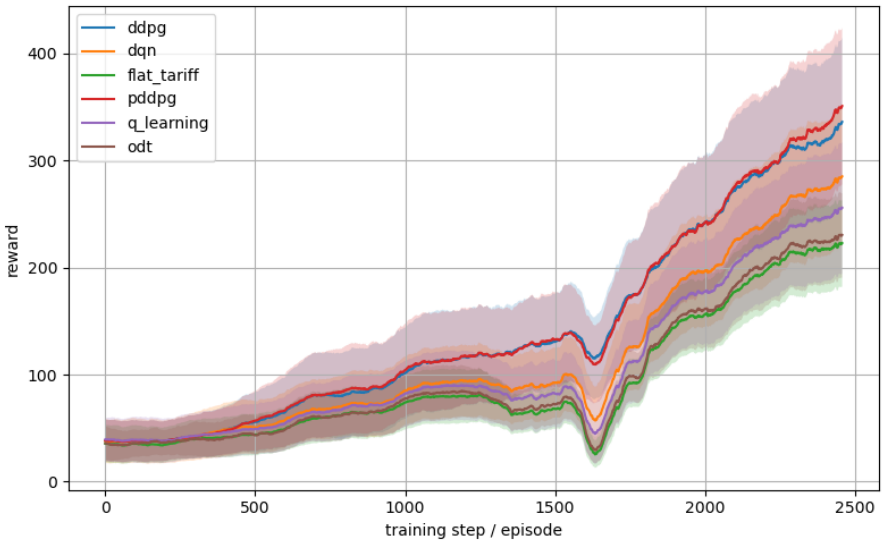

Figure 13: Training curve comparison of different learning-based methods for 2018-2024, using only $w_{\text{rev}} = 1$ against a competitor using a flat tariff of 45 cents

Model	Test mean	Test std
DDPG	376.40	37.31
DQN	321.87	11.11
Flat tariff	248.53	2.69
ODT	259.66	3.60
PDDPG	389.46	20.09
Q-learning	284.24	3.07

Table 9: The result comparison of different learning-based models on the test year 2025, using only $w_{\text{rev}} = 1$ against a competitor using a flat tariff of 45 cents, averaged over 20 runs

7.2.4 Revenue only against RL competitor

Figure 14 shows all the training curves for all pricing strategies when the goal is only revenue optimization and the competitor is able to adapt its pricing strategy using a DQN to maximize their own revenue.
All learning-based methods show an upward trajectory again, as in Figure 13. This time, the overall gaps between methods seem to be higher. The gap between the PDDPG and DDPG is wider. There is a clear gap now between the online decision transformer and the flat tariff baseline, which are around equal levels in the previous setup. The gap between the DQN and the DDPG also seems wider, further showing that the policy-gradient methods perform relatively better in the revenue only setup. The overall relative performance remains consistent as in Figure 13 as the ranking is still: PDDPG, DDPG, DQN, Q-learning for the top 4 methods.

Table 10 shows the test performance for the year 2025. PDDPG achieves the highest mean test reward (457.47), followed by DDPG (434.28) and DQN (370.85). All of these values are higher than in Table 9, while the relative ranking seems to be consistent for the better performing methods. As these performance values are higher, the gap between methods has also widened. For some methods, the standard deviation is also higher, while the standard deviation for some methods has only slightly changed. Notably, the standard deviation for the DDPG changes only by 1.51. The other methods all have higher standard deviations. Even though the gap has widened in this setup compared to the previous experiment, Figure 18 shows no difference in statistical significance between the top performing values compared to Figure 17.
As in the previous experiments, the values in Table 10 correspond directly to monetary value. However, these results are the mean per day. The total overview of revenue and other metrics can be found in Section 7.2.5.

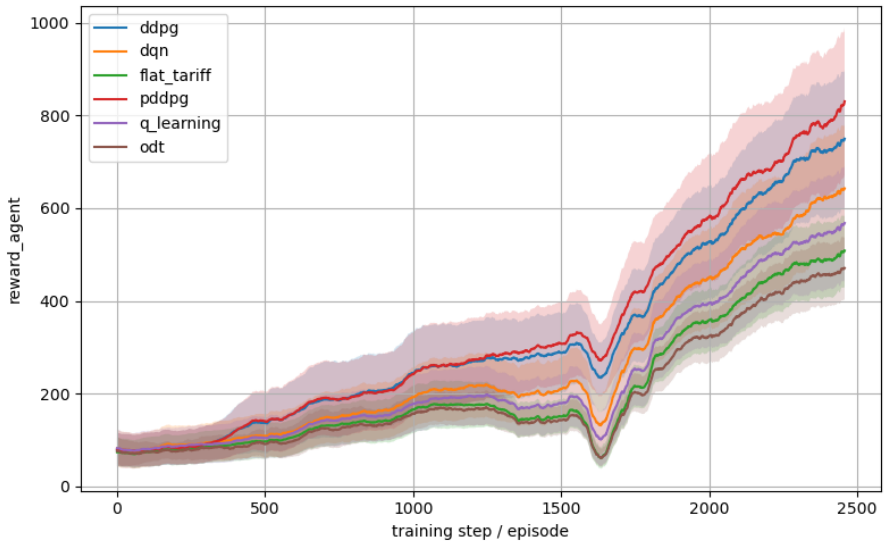

Figure 14: Training curve comparison of different learning-based methods for 2018-2024, using only $w_{\text{rev}} = 1$ against a competitor using a DQN to maximize its revenue, averaged over 20 runs

Model	Test mean	Test std
DDPG	434.28	35.80
DQN	370.85	27.43
Flat tariff	268.54	12.86
ODT	284.44	5.99
PDDPG	457.47	42.85
Q-learning	311.89	17.34

Table 10: The result comparison of different learning-based models on the test year 2025, using only $w_{\text{rev}} = 1$ against a competitor using a DQN to optimize their revenue

7.2.5 Other test year metrics

This section shows all other metrics like revenue, sessions and MWh sold for both the agent and the competitor. These metrics are shown for the model that has the highest test mean

for each specific environment. That means that the DDPG result metrics are shown for the experiments in Sections 7.2.1 and 7.2.2 and the PDDPG result metrics for Sections 7.2.3 and 7.2.4. All values shown in this section are year total values averaged over 20 independent runs. Table 11 shows the baseline case in which the agent and the competitor use a flat tariff of 45 cents per kWh. This scenario represents a situation where no learning or dynamic pricing is present. The agent and the competitor show comparable values in revenue, delivered energy and number of charging sessions.
The total energy and congestion energy are relatively stable across runs, as indicated by the low standard deviations. The agent seems to average more sessions, while still selling less volume than the competitor. This is also represented in the split of congestion energy, leading to a total of 192.40 MWh charged during congestion hours over the test year.

Metric	Mean ± std
Agent Revenue (EUR)	90,712.66 ± 980.56
Competitor Revenue (EUR)	94,263.78 ± 985.42
Agent Volume (MWh)	252.54 ± 2.73
Competitor Volume (MWh)	262.20 ± 2.69
Agent Sessions	14,824 ± 110
Competitor Sessions	14,234 ± 112
Agent congestion energy (MWh)	92.56 ± 1.32
Competitor congestion energy (MWh)	99.84 ± 1.17
Total congestion energy (MWh)	192.40 ± 1.66

Table 11: Metrics other than reward over the test year of 2025 using a flat tariff of 45 cents for both the agent and the competitor

Table 12 shows the metrics for the DDPG agent in the environment with a static competitor. Its reward is based upon the weights described in Section 5.2.10. Compared to the baseline, the agent's total revenue increases, while total energy delivered decreased. This implies that the prices are set on average higher than the baseline of 45 cents. Due to this price increase, the competitor sees an increase in sessions and volumes sold.
The agent's congestion energy is lower than in the baseline, while the competitor's congestion energy increases. The total amount of congestion energy is slightly lower than in the baseline, but most of the congestion energy of the agent seems to be redistributed to the competitor. Variability across runs is notably higher overall than in the baseline.
Table 13 shows the corresponding metrics when the competitor is allowed to learn using a DQN. In this setting, both operators show higher revenues compared to the static-competitor case. The revenue, volume and sessions seem closer to the baseline than in the static-competitor DDPG setup.
Compared to Table 12, standard deviations for most metrics are larger. The slight decrease in total congestion energy level is now close to the baseline again and the slight decrease seen in Table 12 is gone. Notably, the total congestion energy is even lower than in the previous setup, even though one of the operators is purely focused on maximizing revenue and does not optimize on reducing congestion energy. However, it is also important to note that the standard deviation of the total congestion energy here is significantly higher than in the previous setups.
Table 14 shows the metrics for the PDDPG agent with a static competitor. The agent here has the task to maximize its revenue. Notably the revenue in this setup is lower than in Table 13, while that agent is not revenue only focused. Delivered energy and amount of sessions are

Metric	Mean ± std
Agent Revenue (EUR)	118,210.51 ± 11,596.96
Competitor Revenue (EUR)	99,147.13 ± 5,554.24
Agent Volume (MWh)	238.60 ± 13.32
Competitor Volume (MWh)	276.04 ± 15.35
Agent Sessions	14,084 ± 783
Competitor Sessions	14,978 ± 835
Agent congestion energy (MWh)	85.77 ± 7.75
Competitor congestion energy (MWh)	103.43 ± 11.59
Total congestion energy (MWh)	189.20 ± 6.25

Table 12: Metrics other than reward over the test year of 2025 using a DDPG given the setup from Section 7.2.1, which has a static competitor

Metric	Mean ± std
Agent Revenue (EUR)	134,521.50 ± 20,267.81
Competitor Revenue (EUR)	126,817.81 ± 15,967.64
Agent Volume (MWh)	249.17 ± 23.21
Competitor Volume (MWh)	263.33 ± 26.15
Agent Sessions	14,766 ± 1,225
Competitor Sessions	14,013 ± 1,218
Agent congestion energy (MWh)	93.87 ± 11.60
Competitor congestion energy (MWh)	92.26 ± 12.93
Total congestion energy (MWh)	186.13 ± 12.37

Table 13: Metrics other than reward over the test year of 2025 using a DDPG given the setup from Section 7.2.2, which allows the competitor to use a DQN to maximize their revenue

the lowest in this setup and are redistributed towards the competitor as they see the highest values in these metrics as well compared to previous setups.
The congestion energy for the agent is reduced significantly compared to the baseline, while the competitor congestion energy is increased. The total amount of congestion energy remains close to the baseline. Again, this implies that the charging demand during congestion hours is redistributed from the agent towards the competitor.
The variability across runs is comparable to the DDPG experiments. When comparing the agent values for revenue, with Table 12, it is notable that the standard deviations are slightly lower. The use of experience replay in the PDDPG agent and the focus on maximizing revenue might be an explanation for this decrease in variability. This revenue focused setup also shows the highest total congestion energy in all setups until now.
Finally, Table 15 shows the metrics for the PDDPG agent when the competitor also uses a learning-based strategy. In this configuration, both the agent and the competitor achieve their highest revenue values.
The competitor seems to still have the higher volume and amount of sessions compared to the agent, while still keeping a similar amount of revenue compared to the agent. Congestion energy is also shifted more towards the competitor, which is consistent with previous setups. Interestingly, the total amount of congestion energy is slightly lower when compared to Table

Metric	Mean ± std
Agent Revenue (EUR)	142,151.73 ± 7,331.10
Competitor Revenue (EUR)	113,225.21 ± 5,699.65
Agent Volume (MWh)	204.02 ± 14.20
Competitor Volume (MWh)	314.40 ± 16.23
Agent Sessions	12,031 ± 838
Competitor Sessions	17,128 ± 875
Agent congestion energy (MWh)	76.68 ± 9.08
Competitor congestion energy (MWh)	116.99 ± 13.12
Total congestion energy (MWh)	193.67 ± 6.76

Table 14: Metrics other than reward over the test year of 2025 using a PDDPG to maximize revenue, which has a static competitor

11 in a setup where all players only focus on maximizing revenue, albeit that this setup has a far higher standard deviation.

Metric	Mean ± std
Agent Revenue (EUR)	166,975.81 ± 15,640.20
Competitor Revenue (EUR)	154,371.75 ± 18,313.39
Agent Volume (MWh)	213.15 ± 16.34
Competitor Volume (MWh)	302.81 ± 29.11
Agent Sessions	12,497 ± 1,013
Competitor Sessions	16,209 ± 1,338
Agent congestion energy (MWh)	75.76 ± 8.83
Competitor congestion energy (MWh)	116.19 ± 15.24
Total congestion energy (MWh)	191.95 ± 12.55

Table 15: Metrics other than reward over the test year of 2025 using a PDDPG to maximize revenue, which allows the competitor to use a DQN to maximize their revenue

Across all of these tables, the total congestion energy varies only within a small range. However, revenue, volume and session allocation can differ between the agent and competitors. As can be seen in the baseline with similar prices in Table 11, all these metrics start out comparatively balanced.

Taken together, the reinforcement learning results demonstrate that policy-gradient methods consistently outperform value-based approaches in the dynamic pricing task, which is consistent with the literature [45]. DDPG and PDDPG achieve higher test-year rewards and revenues than Q-learning and DQN across all evaluated scenarios.

The performance gap between policy-gradient and value-based methods is consistent across evaluated scenarios, rather than scenario-specific. The DDPG and PDDPG achieve higher test rewards and revenues and also seem to generally converge faster when comparing their learning curves to those of Q-learning and DQN. Using a multi-objective reward function, the DDPG generally outperforms the other methods, albeit not statistically significantly. In a revenue optimizing setting, the PDDPG consistently achieved the highest reward, also not statistically significantly. Appendix A shows no statistical significance in the performance of the DDPG or PDDPG using a Friedman test followed by a Nemenyi post-hoc analysis against a static competitor when compared to each other. Furthermore, in every tested setup the DQN

performs statistically equivalently to the best performing method present. Using the same test, the value-based methods always show statistical equivalence. Nonetheless, the reinforcement learning methods do consistently and statistically outperform the flat tariff baseline.

7.3 Transformer model comparison results

This subsection compares the performance of transformer-based models in the dynamic pricing environment. The objective of this comparison is not to evaluate learning dynamics, but to evaluate whether the selected transformer models can generate competitive day-ahead pricing strategies under identical operational constraints.
The time-series foundation models are used in a zero-shot manner through closed-loop continuation and are unable to observe rewards or demand. Their only input is the previous set of actions given by a DDPG or PDDPG agent. The decision transformer is trained on historical trajectories and reproduce behaviour learned offline. All models are evaluated on the test year 2025 using the same metrics as the reinforcement learning experiments. No reinforcement learning competitor is trained in this experiment.
Table 16 shows the performance of the transformer-based models on the test year 2025 under the multi-objective reward setting against a static competitor. The results are compared to the best-performing reinforcement learning baseline, the DDPG from Table 7.
Among the transformer-based methods, the decision transformer achieves the highest mean performance and the lowest variance. Important to note is that all these models have received previous actions made by the DDPG. The decision transformer is trained on these actions and their returns and shows strong similarity to the performance of the DDPG, but does not achieve a higher test mean. It does show a more stable result than the zero-shot foundation models. Chronos achieves the lowest performance and stability when compared to the other learning-based methods in Table 7. TimesFM shows performance similar to the PDDPG and Q-learning in the previous experiment. However, TimesFM shows a higher standard deviation than the previous highest value, achieved by PDDPG, of 8.82.
TimesFM outperforms Chronos in both average reward and stability. However, the classical decision transformer outperforms both foundation models in mean and standard deviation. The best performance, shown by the DDPG, highlights that there is still a large advantage in interacting with this environment. This does not mean that transformer-based methods perform worse than learning-based methods. However, it does show that the context of the environment given to the learning-based methods is important. Without it, using foundation models in a zero-shot manner in this setting could lead to a worse performance than methods that optimize their objectives through interaction.

Model	Test mean	Test std
Chronos	-231.52	22.48
TimesFM	-213.63	13.33
DT	-204.92	9.90
DDPG	-201.01	5.69

Table 16: The result comparison of transformer-based models on the test year 2025 using the weights described in Section 5.2.10 against a static competitor, along with the best performing learning-based method: DDPG

Table 17 shows the performance of the transformer-based models on the test year 2025 in a

revenue only setting, against a static competitor. Under this simplified objective, the decision transformer achieves a mean test score that is closely aligned with the PDDPG baseline, which has the best test mean in the previous experiment. The standard deviation is slightly lower than the PDDPG. Again, this shows that the decision transformer is strong in learning from actions previously given by the PDDPG, but is unable to outperform it. However, it does perform similarly to the DDPG from Table 13 with a lower standard deviation.
The zero-shot foundation models do not benefit from this simplified setting. Both TimesFM and Chronos have a higher standard deviation than the previous highest value of 23.30 by the DDPG. TimesFM achieves a test mean between Q-learning and the DQN, while Chronos performs worse than the flat tariff baseline. These results show that in this environment, the zero-shot price continuation based solely on historical price sequences is difficult. Due to the absence of the context, the foundation models could struggle to adapt their forecasts to the stochastic elements in this environment. The interaction of the learning-based models gives them an advantage that is hard for the zero-shot models to adapt to.

Model	Test mean	Test std
Chronos	189.16	173.32
TimesFM	302.16	72.01
DT	373.53	17.85
PDDPG	389.46	20.09

Table 17: The result comparison of transformer-based models on the test year 2025 only optimizing revenue against a static competitor, along with the best performing learning-based method: PDDPG

Overall, the transformer-based models do not outperform the best-performing reinforcement learning agents on the 2025 test year under identical operational constraints. While the decision transformer achieves competitive performance by reproducing previously observed action–return patterns, it does not exceed the test-year performance of the DDPG in the multi-objective setting or the PDDPG in the revenue only setting. The zero-shot foundation models exhibit lower average performance and higher variance.
The decision transformer closely matches the performance of the best reinforcement learning agents. However, it does not outperform the DDPG or PDDPG baselines. In contrast, the zero-shot time-series foundation models show lower average performance and higher variance, indicating limited robustness in the absence of environmental context and reward feedback. Overall, reinforcement learning methods remain superior for dynamic EV charging price optimization in this environment, while the decision transformer provides a competitive but constrained alternative and the time-series foundation models seem to show lowered performance due to their lack of context when using it in a zero-shot manner.

8 Discussion

This section discusses the results presented in the previous chapter and places them in a broader context. The findings are interpreted and the limitations of the proposed simulation environment are discussed. Additionally, this section discusses possible future work.

8.1 Interpretation of findings

This thesis investigates the effectiveness of reinforcement learning and transformer-based models for dynamic EV charging price optimization in a semi-realistic environment. The results show clear performance differences between model classes and provide insights into the importance of interaction and context in pricing decisions.

8.1.1 Interpretation of environment results

Firstly, the results of the environment experiments do show expected behaviour from electricity demand in the environment. The observed inverse relationship between price levels and both delivered energy and charging sessions seen in Section 7.1.2 confirms that the environment responds consistently to price signals. The results align with the expected behaviour from prior literature. These findings therefore support H_env2. The peak surcharge in Section 7.1.3 leads to a redistribution of charging demand between operators rather than a reduction in total peak-hour energy consumption across the network. While higher surcharges successfully shift demand away from agent-owned stations, they do not decrease aggregate peak-hour charging demand. These results therefore reject H_env3. Lastly, the estimated PED values seen in Section 7.1.4 are negative across all price intervals, which is consistent with the expected behaviour of electricity demand, which Lijesen [32] estimated around -0.05 to -0.6. The magnitude of the elasticity increases with higher price levels, meaning that demand becomes more price-responsive as charging prices rise. This is in line with the expected behaviour of electricity demand and therefore supports H_env4.

8.1.2 Interpretation of learning models results

The reinforcement learning experiments show that the policy-gradient methods consistently outperform value-based approaches in this environment in the tested setups, albeit not always statistically significant. This trend suggests that models capable of continuous action decisions are better suited for this high-dimensional day-ahead pricing problem. The strongest performances belong either to the DDPG or PDDPG. While their variance is consistently higher, their mean reward and generally faster convergence make them a strong choice. The DDPG achieves better results in a multi-objective reward function, while the PDDPG achieves higher results in a revenue only optimizing configuration. The prioritized experience replay, which is the difference between the DDPG and PDDPG could be the explanation for these findings. In a revenue only setup, it could be more viable to prioritize and replay actions that are previously found to receive good rewards. This process could be prone to overfitting, which can also be used to explain the decreased performance in a multi-objective setting, as it would be more difficult to prioritize good actions in such a configuration.
Furthermore, when both operators use reinforcement learning, the environment can be interpreted as a non-cooperative Markov game rather than a single-agent decision problem. In this setting, each operator's outcome depends not only on its own pricing policy, but also on the adaptive behaviour of the competing operator. The results suggest that when both agents learn, the revenue gap between them becomes substantially smaller than in settings where only one operator uses reinforcement learning. This indicates mutual strategic adaptation and a more balanced competitive outcome. Although this should not be interpreted as proof of a formal Nash equilibrium, it is consistent with the idea that repeated interaction between learning agents can produce behaviour that approximates equilibrium-like pricing dynamics.

8.1.3 Interpretation of transformer-based model results

The interaction within this environment seems to be of high importance for the success of a method. Zero-shot time-series foundation models, while generally powerful, perform overall worse than the learning-based methods in this environment. They yield higher variance and lower reward. However, this does not mean that transformer-based methods, or even particularly foundation models, can not perform well for EV dynamic pricing strategies. This thesis shows exploratory performances of these methods without any fine-tuning. These methods might perform significantly better if more research time is spent on their ideal configurations. The classical decision transformer has achieved performance close to the best reinforcement learning baselines by reproducing their previous action-return patterns. As the performance of the online decision transformer shows, these methods are not best used to directly replace classical reinforcement learning algorithms. They are however, capable of being used as a complement and can likely achieve good results when used alongside an already established method. They do not surpass the established reinforcement learning methods in this present work. This shows the importance of interaction in this environment.
Summarising, these findings show that dynamic EV charging pricing strategies are best interpreted as a decision problem. Methods that adapt their decision through interaction with the environment consistently outperform approaches that rely only on historical pattern continuation.

8.2 Implications for dynamic EV charging pricing

An important insight from the environment validation experiments is that static peak surcharges not only redistribute charging demand between operators, but even worsened peak load across the network. A common concern with applying dynamic pricing strategies is that raising prices will make customers leave for a competitor. Not only does this hold true in this environment, but another interesting interaction is shown. Normally, the EV driver can consider delaying their charging by an hour if the next price is more affordable. If the surcharge is too high however, the EV driver is more likely to leave for a competitor than consider delaying. This also reflects long-term behaviour as EV drivers will more likely learn to either charge at a competitor or to not charge at all during peak congestion hours. This finding has implications for real-world pricing policies: operator-level incentives do not necessarily translate into system-level benefits when competitors are present.
In this setup the usage of reinforcement learning agents did show a reduction in total congestion energy to help reduce network-level issues. This is shown in both multi-objective setups against a static competitor and a revenue optimizing one. The total amount of energy during congestion hours is slightly reduced, albeit not by a significant amount. This implies a possibility for further research in dynamic pricing for EV charging to reduce energy delivered in peak demand hours. Therefore, the results do support the use of data-driven pricing mechanisms over static tariff approaches to reduce peak hour charging.
Another important insight is that the revenue of the agents always improves when using a learning approach compared to the flat tariff baseline scenario. In both cases where the competitor and agent use reinforcement learning, the gap between the revenues becomes smaller than in a setup with a static competitor. There seems to be some form of equilibrium achieved between the two, normally adversarial players, where they both result in similar revenue. If the competitor does not use reinforcement learning, there is a 20,000 and 30,000 euro difference when using multi-objective rewards or revenue only respectively. When the competitor is allowed

to use a learning-based method, the difference becomes 8,000 euro. Even if the agent is not trying to maximize revenue and the competitor is in the multi-objective setting, their performance in revenue is close. When looking from a revenue perspective, it is therefore very profitable to be using these data-driven methods.
More broadly, these results show that the pricing mechanisms aimed at mitigating grid congestion cannot be effectively utilized by one operator. When multiple operators compete for the same demand, attempts at congestion mitigation may end up simply redistributing energy rather than reducing it. Monetarily, it is beneficial for each market player to use learning-based decision algorithms, which might not be in the best interest of the EV driver or the network. From a policy perspective, these results show that dynamic pricing schemes may require charge point operators to behave cooperatively instead of adversarially, since it is possible to lower the peak demand charging. Not modelled factors such as regulations and raised grid-awareness might also help to ensure system reliability for both the charge point operator and the EV driver.

8.3 Limitations of the simulation environment

Despite the semi-realistic design of the simulation environment, several limitations must be acknowledged. First, the charging network topology and arrival processes are modelled abstractions and do not fully capture the spatial and temporal complexity of real-world charging behaviour. While the environment shows plausible responses to pricing signals and congestion incentives, it remains a simplified representation of reality. The charging network topology is a simplified model and while it is both possible and plausible to model a charging network this way, there are numerous factors not taken into consideration for this topology.
The placement of a charging station for example is chosen with numerous considerations, which is not reflected in this simplified version. Infrastructure, accessibility and actual geographical data are for instance not represented. The modelling choice for a maximum of three edges per node is also a simplification, as it is possible for a single charge point operator to own a large connected component with no nearby competitors. An example would be a shopping centre where there is only one charge point operator and possibly more than three nearby charging stations. It is also not true that there is always at least one other charging station nearby, as it is possible for there to be none.
Furthermore, the arrivals are based on the synthetic data of the average charging station. In practice, there is a large variety in the arrival and utilization rate between different chargers. Some chargers see very high usage and others very little. This relationship is only modelled in the environment using the size of the connected components and not per individual charger.
Furthermore, the simulation assumes a fixed definition of peak hours, which might not align with actual congestion periods in different regions. It is possible that the usage of dynamic pricing and other data-driven EV technologies, like smart charging, moves the demand peak outside of these hours. These peak hours are not representative of real net congestion, which is only modelled using a simplified maximum amount of energy deliverable by the network. No blackout event caused by net congestion is modelled and no location-based physical congestion. There is no complex design for congestion in this environment, instead there is a simplified hour system.
This environment allows plausible behaviour of EV drivers, but it is not completely realistic yet. Firstly, the assumption that EV drivers have perfect information about the prices for the entire day is not completely realistic. It is possible for EV drivers to become aware of these prices in real life, but they do not have to be aware of the price to start a charging session. Sometimes,

an EV driver only finds out the price of their session in the form of a monthly invoice and only then possibly change their behaviour. B2B drivers are also a large share of EV drivers. The invoices would be sent to their employers and they would likely not be able to reflect on their charging behaviour. Furthermore, numerous more complex driver behaviours are not modelled realistically, like brand loyalty and the urgency of their charge. EV drivers will sometimes be forced or be very willing to charge somewhere depending on their battery percentage.
During the writing of this thesis, the pricing of electricity had been set from hourly to quarterly. The impact of this change is significant, yet the choice to keep the environment hourly remained. The training is done for the years 2018-2024 and at that time there was no quarterly data available yet. The price set by a charging station operator is also not the exact price the EV driver will see, as there are also taxes, fees and other rules placed upon these prices. These are not reflected in this environment.
Finally, there is a limit of two EV drivers being able to charge at the same charging station at the same time. The average charging station in the Netherlands does indeed allow for this, but there are some that might allow less or more EV drivers at the same time. These are public charging stations and there are no fast chargers modelled in this environment. The chargers in this environment will allow a maximum speed of 11 kW, which is a plausible theoretical maximum. In practice, it is impossible to model directly what the charging speed would actually be of an EV. This depends on, for example, the state of their battery, the amount of phases in the car itself or if the charging station is already being used by another EV. There is also no ability to form a queue for a charging station, which in real life could happen. In this environment, if there is no place available nearby, the driver will either leave or try another connected component altogether.

8.4 Implications for real-world deployment

The findings of this thesis suggest that reinforcement learning-based pricing strategies have the potential to improve both network level issues and charge point operator revenue. Deploying such methods in real-world systems requires careful consideration due to regulations. The deployment of learning-based pricing strategies raises practical considerations beyond algorithmic performance. Regulatory constraints may limit the frequency or magnitude of price adjustments, and transparency requirements may restrict the use of highly adaptive pricing mechanisms. Furthermore, customer trust plays a crucial role in the adoption of dynamic pricing, particularly in public charging infrastructure where pricing volatility may be perceived negatively by users. This thesis has highlighted the importance of high-quality real-world data for the training of these learning-based methods. Due to the zero-shot foundation models missing context, it can be assumed that they may be insufficient for operational pricing decisions without further fine-tuning.
Recent changes in the electricity market, such as the transition from hourly to quarterly pricing, show that data-driven decision-making in the EV sector is highly dependent on ongoing improvements. While the proposed framework can be extended to further timelines, the complexity or dimensionality of this project might need to increase in the future.

8.5 Directions for future research

Several directions for future work emerge from this study. Incorporating real-world charging session data from multiple years would allow for a more robust assessment of generalization and

long-term adaptation. Extending the environment to include renewable energy generation and dynamic grid constraints would further improve realism and enable the evaluation of grid-aware pricing strategies.
Furthermore, more varied experiments should be done within this environment. In the Netherlands, it is unusual for there to be only two charge point operators represented in a certain area. More insights could potentially be gained from having more than two players in the environment. Furthermore, more weight configurations could be tried to gain more insight into possible EV driver behaviour and market dynamics. For example, an experiment where both the agent and the competitor try to minimize charging during peak hours.
Overall, future research should focus on increasing environmental realism, improving data availability, and exploring hybrid learning methods that combine the generalization capabilities of large-scale models with the adaptability of reinforcement learning. Such extensions would be necessary to bridge the gap between controlled simulation studies and real-world deployment of dynamic EV charging pricing strategies.

9 Conclusion

The increasing adoption of electric vehicles places growing pressure on public charging infrastructure and local electricity grids. Dynamic pricing is often proposed as a mechanism to manage charging demand and reduce congestion. This thesis investigated the extent to which learning-based pricing strategies can address these challenges by comparing reinforcement learning methods and transformer-based methods in a semi-realistic environment for EV charging.
A simulation framework was developed that captures key aspects of public EV charging. It includes stochastic arrivals, local competition between charge point operators and congestion mitigation objectives. In this environment, multiple learning-based methods and transformer-based models were tested under the same evaluation metrics. The results provide insights into the suitability of different approaches to dynamic EV charging price optimization.
Research Question 1 investigated whether the proposed environment is stable, reproducible and representative enough. The environment validation experiments demonstrate that the simulation responds consistently to pricing signals, show realistic price elasticity of demand and produce reproducible outcomes across repeated runs. Overall, the results confirm that the environment is robust and economically plausible to support evaluation of learning-based pricing strategies.
Research question 2 investigated the extent to which selected reinforcement learning methods differ in their ability to optimize dynamic EV charging prices under semi-realistic constraints. The results show clear and systematic performance differences between value-based and policy-gradient approaches. The DDPG and PDDPG consistently achieve higher performance than Q-learning and DQN in terms of achieved reward, revenue, and learning efficiency, but did not show statistical significance. They did always perform statistically better than a flat tariff baseline. These findings indicate that reinforcement learning is indeed effective for day-ahead pricing problems and the continuous action space of the policy-gradient methods might be more effective in the environment than the discrete DQN and Q-learning methods.
Research question 3 investigated how transformer-based models compare to reinforcement learning methods for dynamic EV charging price optimization. These results show that transformer-based approaches can generate competitive pricing strategies, but do not outperform

reinforcement learning agents under the evaluated conditions. The decision transformer is capable of reproducing the behaviour of strong reinforcement learning policies when trained on their trajectories. However, they are unable to surpass them. Zero-shot time-series foundation models however show lower performance and high variance, which is likely due to their lack of environmental context or feedback. These findings suggest that while transformer-based methods are powerful, and could be even more powerful with fine-tuning instead of zero-shot application, the interaction with the environment plays a strong factor in the effectiveness of pricing decisions in this stochastic and competitive setting.

This thesis reveals structural limitations of static pricing interventions. Peak surcharges primarily redistribute demand across operators rather than reducing aggregate peak load, further strengthening the need for adaptive and context-aware pricing mechanisms. These insights are relevant for charging network operators, grid operators, and policymakers considering dynamic pricing as a tool for congestion management.

The conclusion of this thesis must be interpreted with the assumptions, simplifications and limitations in mind. While the simulation environment is developed to capture key behavioural and economic mechanisms, it lacks a complex model of physical grid constraints, detailed geospatial structure and long-term behaviour adaption. The presented results should be understood as comparative evidence of method performance and incentive effects within a controlled setting rather than as deployment-ready pricing policies.

Usage of AI

This thesis made limited use of ChatGPT by OpenAI to support parts of the writing process. The tool was used for spellchecking, grammar improvement or rephrasing of text. All output generated by the tool has been reviewed and validated by the author.

References

[1] Graziano Abrate, Giovanni Fraquelli, and Giampaolo Viglia. "Dynamic pricing strategies: Evidence from European hotels". In: *International Journal of Hospitality Management* 31.1 (2012), pp. 160–168.

[2] Mohamed H Albadi and Ehab F El-Saadany. "A summary of demand response in electricity markets". In: *Electric power systems research* 78.11 (2008), pp. 1989–1996.

[3] Abdul Fatir Ansari et al. "Chronos: Learning the language of time series". In: *arXiv preprint arXiv:2403.07815* (2024).

[4] ANWB. *Hoe lang duurt het opladen van een elektrische auto?* 2025. URL: `https://www.anwb.nl/auto/elektrisch-rijden/opladen/hoe-lang-duurt-opladen-elektrische-auto` (visited on 01/11/2026).

[5] ANWB. *Wat kost het opladen van een elektrische auto?* 2025. URL: `https://www.anwb.nl/auto/elektrisch-rijden/kosten/wat-kost-elektrisch-laden` (visited on 01/11/2026).

[6] Victor F Araman and René Caldentey. "Revenue management with incomplete demand information". In: *Encyclopedia of Operations Research. Wiley* (2011).

[7] Daehyun Ban, George Michailidis, and Michael Devetsikiotis. "Demand response control for PHEV charging stations by dynamic price adjustments". In: *2012 IEEE PES Innovative Smart Grid Technologies (ISGT)*. IEEE. 2012, pp. 1–8.

[8] Dimitris Bertsimas and Georgia Perakis. "Dynamic pricing: A learning approach". In: *Mathematical and computational models for congestion charging* (2006), pp. 45–79.

[9] Leonardo De A Bitencourt et al. "Optimal EV charging and discharging control considering dynamic pricing". In: *2017 IEEE Manchester PowerTech*. IEEE. 2017, pp. 1–6.

[10] Severin Borenstein. "The long-run efficiency of real-time electricity pricing". In: *The Energy Journal* 26.3 (2005), pp. 93–116.

[11] Dušan Božič and Miloš Pantoš. "Impact of electric-drive vehicles on power system reliability". In: *Energy* 83 (2015), pp. 511–520.

[12] Lili Chen et al. "Decision transformer: Reinforcement learning via sequence modeling". In: *Advances in neural information processing systems* 34 (2021), pp. 15084–15097.

[13] Qin Chen and Komla Agbenyo Folly. "Application of artificial intelligence for EV charging and discharging scheduling and dynamic pricing: A review". In: *Energies* 16.1 (2022), p. 146.

[14] Abhimanyu Das et al. "A decoder-only foundation model for time-series forecasting". In: *Forty-first International Conference on Machine Learning*. 2024.

[15] Janez Demšar. "Statistical comparisons of classifiers over multiple data sets". In: *Journal of Machine learning research* 7.Jan (2006), pp. 1–30.

[16] Arnoud V Den Boer. "Dynamic pricing and learning: historical origins, current research, and new directions". In: *Surveys in operations research and management science* 20.1 (2015), pp. 1–18.

[17] Goutam Dutta and Krishnendranath Mitra. "A literature review on dynamic pricing of electricity". In: *Journal of the Operational Research Society* 68.10 (2017), pp. 1131–1145.

[18] ElaadNL. *Charging data and simulation profiles*. 2025. URL: `https://charging.elaad.nl/` (visited on 01/10/2026).

[19] Ahmad Faruqui and Sanem Sergici. "Household response to dynamic pricing of electricity: a survey of 15 experiments". In: *Journal of regulatory Economics* 38.2 (2010), pp. 193–225.

[20] Ahmad Faruqui, Sanem Sergici, and Ahmed Sharif. "The impact of informational feedback on energy consumption—A survey of the experimental evidence". In: *Energy* 35.4 (2010), pp. 1598–1608.

[21] Peter J Huber. "Robust estimation of a location parameter". In: *Breakthroughs in statistics: Methodology and distribution*. Springer, 1992, pp. 492–518.

[22] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

[23] International Energy Agency. *The Netherlands 2024*. 2024. URL: `https://iea.blob.core.windows.net/assets/2b729152-456e-43ed-bd9b-ecff5ed86c13/TheNetherlands2024.pdf`.

[24] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. "Reinforcement learning: A survey". In: *Journal of artificial intelligence research* 4 (1996), pp. 237–285.

[25] Georgios Karatzinis et al. "Chargym: An ev charging station model for controller benchmarking". In: *IFIP International Conference on Artificial Intelligence Applications and Innovations*. Springer. 2022, pp. 241–252.

[26] Byung-Gook Kim et al. "Dynamic pricing and energy consumption scheduling with reinforcement learning". In: *IEEE Transactions on smart grid* 7.5 (2015), pp. 2187–2198.

[27] Diederik P Kingma. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[28] John Frank Charles Kingman. *Poisson processes*. Vol. 3. Clarendon Press, 1992.

[29] Haoxuan Kuang et al. "Unraveling the effect of electricity price on electric vehicle charging behavior: A case study in Shenzhen, China". In: *Sustainable Cities and Society* 115 (2024), p. 105836.

[30] Sangyoon Lee and Dae-Hyun Choi. "Dynamic pricing and energy management for profit maximization in multiple smart electric vehicle charging stations: A privacy-preserving deep reinforcement learning approach". In: *Applied Energy* 304 (2021), p. 117754.

[31] Yuxuan Liang et al. "Foundation models for time series analysis: A tutorial and survey". In: *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*. 2024, pp. 6555–6565.

[32] Mark G Lijesen. "The real-time price elasticity of electricity". In: *Energy economics* 29.2 (2007), pp. 249–258.

[33] Timothy P Lillicrap et al. "Continuous control with deep reinforcement learning". In: *arXiv preprint arXiv:1509.02971* (2015).

[34] Michael L Littman. "Markov games as a framework for multi-agent reinforcement learning". In: *Machine learning proceedings 1994*. Elsevier, 1994, pp. 157–163.

[35] Zhigang Lu et al. "Modelling dynamic demand response for plug-in hybrid electric vehicles based on real-time charging pricing". In: *IET Generation, Transmission & Distribution* 11.1 (2017), pp. 228–235.

[36] Chao Luo, Yih-Fang Huang, and Vijay Gupta. "Dynamic pricing and energy management strategy for EV charging stations under uncertainties". In: *arXiv preprint arXiv:1801.02783* (2018).

[37] Monowar Mahmud et al. "Integrating demand forecasting and deep reinforcement learning for real-time electric vehicle charging price optimization". In: *Utilities Policy* 96 (2025), p. 102038.

[38] Volodymyr Mnih et al. "Human-level control through deep reinforcement learning". In: *nature* 518.7540 (2015), pp. 529–533.

[39] Valeh Moghaddam et al. "An online reinforcement learning approach for dynamic pricing of electric vehicle charging stations". In: *IEEE Access* 8 (2020), pp. 130305–130313.

[40] Zeinab Moghaddam et al. "Smart charging strategy for electric vehicle charging stations". In: *IEEE Transactions on transportation electrification* 4.1 (2017), pp. 76–88.

[41] Rijksdienst voor Ondernemend Nederland. *Elektrische personenauto's in Nederland*. 2025. URL: `https://duurzamemobiliteit.databank.nl/mosaic/nl-nl/elektrisch-vervoer/personenauto-s` (visited on 01/10/2026).

[42] Stavros Orfanoudakis et al. "Ev2gym: A flexible v2g simulator for ev smart charging research and benchmarking". In: *IEEE Transactions on Intelligent Transportation Systems* (2024).

[43] Zsuzsanna Pató. *Gridlock in the Netherlands*. 2024. URL: `https://www.raponline.org/wp-content/uploads/2024/01/RAP-Pato-Netherlands-gridlock-2024.pdf`.

[44] Martin L Puterman. "Markov decision processes". In: *Handbooks in operations research and management science* 2 (1990), pp. 331–434.

[45] Dawei Qiu et al. "A deep reinforcement learning method for pricing electric vehicles with discrete charging levels". In: *IEEE Transactions on Industry Applications* 56.5 (2020), pp. 5901–5912.

[46] Rijksoverheid. *Overheid herhaalt campagne om stroomverbruik tijdens piekuren te verminderen*. 2025. URL: `https://www.rijksoverheid.nl/actueel/nieuws/2025/12/15/overheid-herhaalt-campagne-om-stroomverbruik-tijdens-piekuren-te-verminderen` (visited on 01/11/2026).

[47] Shengnan Shao et al. "Impact of TOU rates on distribution load shapes in a smart grid with PHEV penetration". In: *IEEE PES T&D 2010*. IEEE. 2010, pp. 1–6.

[48] Matthijs TJ Spaan. "Partially observable Markov decision processes". In: *Reinforcement learning: State-of-the-art*. Springer, 2012, pp. 387–414.

[49] Kenneth E Train. *Discrete choice methods with simulation*. Cambridge university press, 2009.

[50] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).

[51] José R Vázquez-Canteli et al. “Citylearn v1. 0: An openai gym environment for demand response with deep reinforcement learning”. In: *Proceedings of the 6th ACM international conference on systems for energy-efficient buildings, cities, and transportation*. 2019, pp. 356–357.

[52] Shiqian Wang et al. “EV charging behavior analysis and load prediction via order data of charging stations”. In: *Sustainability* 17.5 (2025), p. 1807.

[53] Christopher JCH Watkins and Peter Dayan. “Q-learning”. In: *Machine learning* 8.3 (1992), pp. 279–292.

[54] Peng Xu et al. “Dynamic pricing at electric vehicle charging stations for queueing delay reduction”. In: *2017 IEEE 37th international conference on distributed computing systems (ICDCS)*. IEEE. 2017, pp. 2565–2566.

[55] Runyao Yu et al. “Pricefm: Foundation model for probabilistic electricity price forecasting”. In: *arXiv preprint arXiv:2508.04875* (2025).

[56] Jeff Zethmayr and David Kolata. “Charge for less: An analysis of hourly electricity pricing for electric vehicles”. In: *World Electric Vehicle Journal* 10.1 (2019), p. 6.

[57] Qinqing Zheng, Amy Zhang, and Aditya Grover. “Online decision transformer”. In: *international conference on machine learning*. PMLR. 2022, pp. 27042–27059.

A Critical difference plots

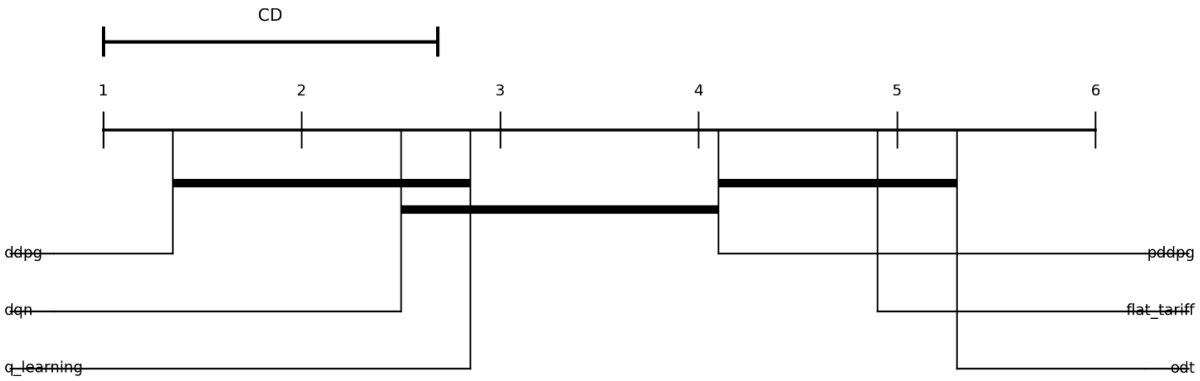

Figure 15: Critical Difference plot that shows the statistical equivalence between methods in the setup where the multi-objective reward function is used against a static price competitor

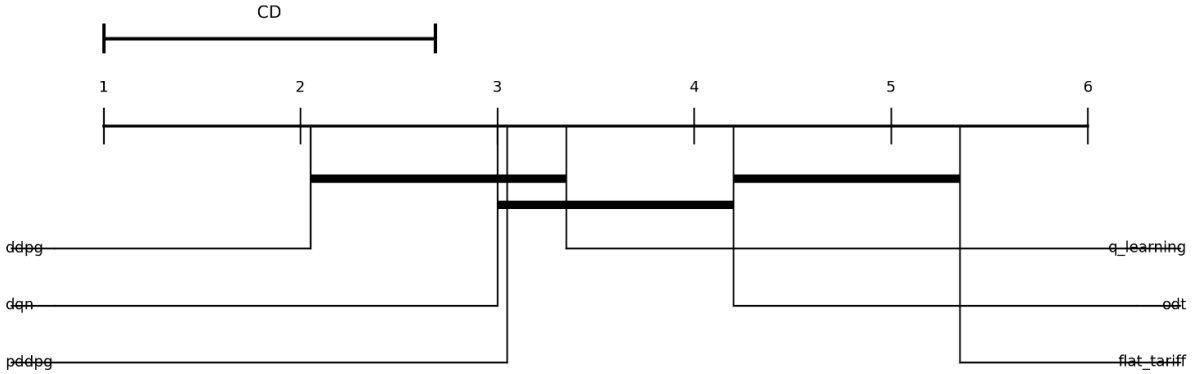

Figure 16: Critical Difference plot that shows the statistical equivalence between methods in the setup where the multi-objective reward function is used against a competitor using a DQN to maximize their revenue

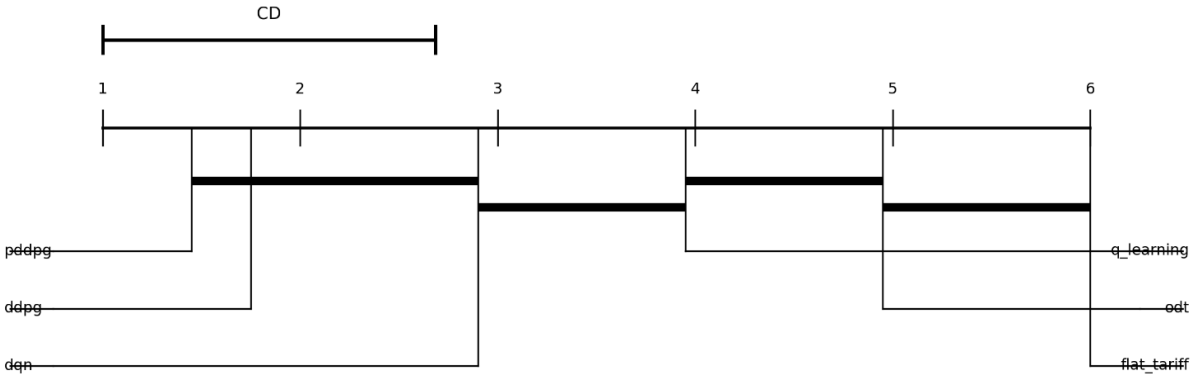

Figure 17: Critical Difference plot that shows the statistical equivalence between methods in the setup where only the revenue is optimized against a static price competitor

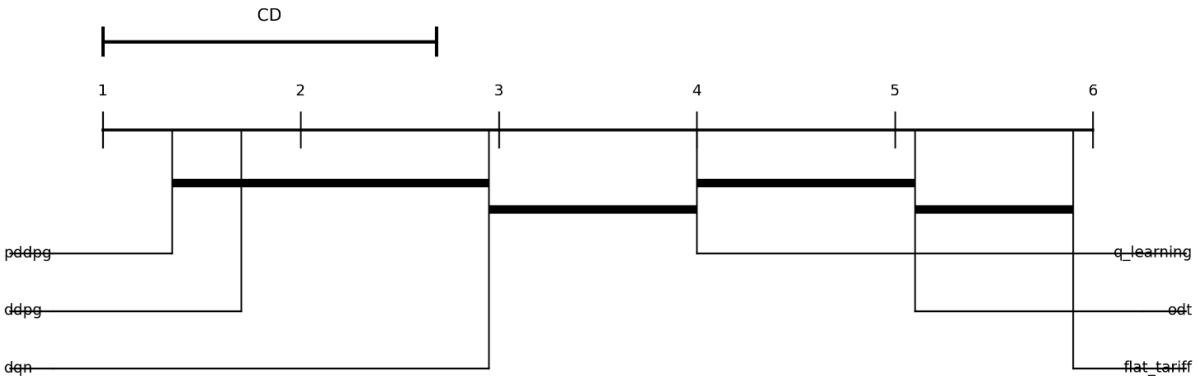

Figure 18: Critical Difference plot that shows the statistical equivalence between methods in the setup where only the revenue is optimized against a static price competitor