Efficient Trading of Aggregate Bidirectional EV Charging Flexibility with Reinforcement Learning

Anonymous Author(s)

ABSTRACT

We study a virtual power plant (VPP) that trades the bidirectional charging flexibility of privately owned plug-in electric vehicles (EVs) in a real-time electricity market to maximize its profit. To incentivize EVs to allow bidirectional charging, we design incentive-compatible, variable-term contracts between the VPP and EVs. Through deliberate aggregation of the energy storage capacity of individual EVs, we learn a reinforcement learning (RL) policy to efficiently trade the flexibility, independent of the number of accepted contracts and connected EVs. The proposed aggregation method ensures the satisfaction of individual EV charging requirements by constraining the optimal action returned by the RL policy within certain bounds. We then develop a disaggregation scheme to allocate power to bidirectional chargers in a proportionally fair manner, given the total amount of energy traded in the market. Evaluation on a real-world dataset demonstrates robust performance of the proposed method despite uncertainties in electricity prices and shifts in the distribution of EV mobility.

CCS CONCEPTS

• **Theory of computation → Algorithmic game theory and mechanism design**; **Mathematical optimization**.

KEYWORDS

Contract Theory, Scheduling, Reinforcement Learning

ACM Reference Format:

Anonymous Author(s). 2024. Efficient Trading of Aggregate Bidirectional EV Charging Flexibility with Reinforcement Learning. In *The 15th ACM International Conference on Future and Sustainable Energy Systems (e-Energy '24), June 4–7, 2024, Singapore.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3575813.3597353

1 INTRODUCTION

Climate change has increased the frequency and intensity of extreme weather events, such as heat waves, wildfires, hurricanes, ice storms, and floods. In recent years, some of these events caused a grid emergency and left millions without power for several hours [4] due to loss of generation capacity, spiking demand for electricity, or weather-related impacts on transmission and distribution systems. Virtual power plants (VPP) – networks of distributed energy resources that are aggregated and controlled to serve the grid – could strengthen the resilience and reliability of the grid in the face of extreme weather events [15]. When these resources are owned by individual customers rather than power utilities, the VPP offers a low-cost alternative to infrastructure upgrades that would otherwise be necessary to improve grid reliability.

An emerging type of VPP aggregates privately owned plug-in electric vehicles (EVs) connecting to residential or public chargers to replenish their battery. This VPP can take advantage of three kinds of flexibility offered by EV chargers to participate in one or multiple electricity markets. First, chargers can regulate their power within certain bounds allowing the charging demand to be shaped. Second, EVs typically remain connected to the charger longer than is needed for their battery to fully charge. This makes it possible to shift the charging demand in time. Finally, as bidirectional chargers with vehicle-to-grid (V2G) functionality become available on the market [1], the EV battery can be discharged for some time before it is charged to the desired state-of-charge (SOC). Compared to VPPs that control a fleet of EVs owned by a company or city [31], e.g. electric taxis or buses, this VPP is capable of offering sizable flexibility to the grid because it can potentially control thousands of privately owned EVs in a large area.

Despite the vast potential of this VPP, ensuring its efficient operation is extremely challenging, especially as the number and diversity of pooled resources increases. This is mainly due to two reasons. First, incentivizing privately owned EVs to allow their battery to be charged and discharged at variable rates is difficult because of individual differences in appraising flexibility and battery degradation costs, and concerns about whether their energy demand would be fulfilled before departure (feasibility concern) and their battery would be discharged by the same amount as others who received the same incentive (fairness concern). Second, volatile prices and stochastic EV mobility make it difficult to guarantee that the available flexibility can be traded efficiently in an electricity market. To address these challenges, Rahman et al. [32] design fixed-term V2G contracts that are offered to EVs upon arrival at the charging station, and develop an online scheduling algorithm to trade their charging flexibility in the imbalance market given knowledge of the accepted contracts and price forecasts. However, fixed-term contracts are too strict, which either prevents many electric vehicles from participating in the VPP (when the contract term is too long) or does not fully utilize their charging flexibility (when the contract term is too short). Additionally, in the real world, price forecasts for several hours in the future are inaccurate, so flexibility cannot be optimally managed by solving a deterministic optimization problem.

We design a set of incentive compatible, variable-term V2G contracts that optimize the expected utility of the VPP. These contracts are offered to each EV as soon as it connects to the charger, allowing it to choose the contract that is best for it. Each contract signifies

e-Energy '24, June 4–7, 2024, Singapore

ACM ISBN 979-8-4007-0032-3/23/06…$15.00
https://doi.org/10.1145/3575813.3597353

the VPP's commitment to EV owners that it will pay them a certain amount of money for discharging a certain amount of energy from their battery over the contract term. The VPP then uses a reinforcement learning (RL) policy to buy electricity to supply the charging demand of all EVs and effectively trade the flexibility in real-time.[1] Since the available flexibility depends on the number of EVs that have accepted a contract, making decisions for each EV independently increases the dimension of the action space, making it difficult to (a) learn a good policy when the number of EVs varies in the dataset, and (b) ensure fairness among EVs that accepted the same V2G contract. To overcome these challenges, our flexibility management approach involves aggregating the energy storage capacity of all EVs into a *virtual battery*, real-time scheduling of the virtual battery operation and accordingly trading flexibility in aggregate in a real-time market, and finally disaggregating the (dis)charge power of the virtual battery into the (dis)charge power of individual EVs, while ensuring fairness and feasibility of these schedules. We make four contributions in this work:

- To incentivize privately-owned EVs to join the coalition, i.e. the VPP, we design self-revealing and variable-term V2G contracts based on the principal-agent model [9] and by extending the agent type to two dimensions. Using realistic parameters for the utility function of the VPP and EV owners, we get nine distinct contracts with three different terms.
- We learn a reinforcement learning policy for trading energy every hour in an electricity market using the single-agent soft actor-critic algorithm that respects time-varying constraints on the action. We show that these constraints can be efficiently calculated by aggregating the laxity and contract-related constraints of individual EV owners. As a result, the learned policy makes reasonable trading decisions irrespective of the number of EVs controlled by the VPP and enforces charging deadlines and terms of accepted contracts.
- We draw a parallel between flexibility disaggregation and resource allocation, as both of them involve self-interested parties, and borrow the notion of proportional fairness from resource allocation to design a fair and efficient flexibility disaggregation algorithm. We compare this algorithm with priority-based disaggregation algorithms that incorporate the notion of laxity.
- We evaluate our flexibility management approach using real data from a network of public charging stations and prices from an imbalance market. Our result reveals that, despite distribution shifts, noisy forecasts, and changes in the number of connected EVs, the learned policy achieves comparable performance with offline optimization problems that receive the same forecasts.

2 RELATED WORK

2.1 VPPs and Flexibility Management

VPPs that combine the flexibility of multiple resources of the same or different types and subsequently trade in the electricity market(s) are extensively studied in the literature [16, 31, 32, 41, 42]. This includes VPPs that aggregate wind and solar generating plants, battery energy storage systems, and EVs. An approach specific to the flexibility in EVs is presented in [34], where unidirectional flexibility is defined in terms of connection time and energy delivered. This flexibility can be managed to flatten the load on the grid or to maximize utilization of renewables. Furthermore, Schlund et al. [36] propose a methodology to aggregate EV flexibility with unidirectional charging to guarantee the availability of bidirectional flexibility over a fixed time horizon. Vandael et. al.[40] propose trading EV flexibility over a finite horizon using dynamic programming. Danner et. al. [14] solve the optimal scheduling of a stationary battery and EVs (without V2G) under various forecast conditions. The flexibility requests are then accommodated by adjusting charging schedules using an iterative algorithm. In follow-on work [13], a genetic algorithm that considers a *fairness index* is proposed. However, unlike our work, they do not consider incentives to EVs and how they affect the feasibility of schedules.

FlexOffers (FOs) [28] define a framework for describing and managing flexibility from heterogeneous resources. The work presented in [37] describes the interaction of different market players using FOs, and in [38] methods for aggregating and disaggregating flexibility while trading in an electricity market are proposed. Moreover, Lilliu et al.[26] expand the definition of FOs by incorporating additional constraints and bidirectional charging. While an individual EV's bidirectional flexibility presented in this work can be expressed as modified FOs, our aggregation schemes are incompatible with FOs as we are interested in preserving the exact flexibility limits through aggregation and disaggregation, while FOs deal with approximate methods.

We are concerned with designing a disaggregation scheme that considers *fairness*. We base our approach on seminal work in resource allocation [24, 45], which introduces the notion of *proportional fairness* and elucidate its properties for elastic traffic in computer networks. We compare our *fair* disaggregation approach with a priority based allocation that was previously adopted in [39].

2.2 Reinforcement Learning for EV Charging

A large number of studies use RL to find an optimal charging schedule for EVs. When there are multiple EVs to control and their number is not fixed, their charging can be scheduled either using a multi-agent reinforcement learning framework [6, 42] or a single-agent reinforcement learning framework that receives the aggregate state representation and produces an aggregate action that can be later disaggregated into the charge or discharge action of every EV. We restrict our literature review to the latter category because of it scales better, does not require information sharing or collaboration among EVs, and allows enforcement of coupling constraints (e.g. for fairness) at the disaggregation stage. Vandael et. al. [39] use a single RL agent to lay out a day-ahead consumption plan, then use priority-based dispatch to disaggregate the action. However, the disaggregated action is not guaranteed to be feasible, so it is clipped at the individual level. Li et. al. [25] develop an aggregator-operator scheme that combines RL with predictive control for discrete binary charging actions by calculating multiple trajectories. Sadeghianpour et. al. [33] propose a streamlined binning approach for unidirectional charging only. Another binning approach is taken by Alshehhi et. al. [7] to create a scalable

[1] By buying more/less energy, the VPP offers negative/positive flexibility to the grid.

representation of EV fleets in order to train a deep neural network to solve a combinatorial optimization problem. Our approach handles charging and V2G contract constraints by aggregating them from an individual level and learning an aggregate action that is guaranteed to be feasible at the individual level.

Learning policies that are guaranteed to produce actions within a safe or feasible set is an active area of research. Chen et. al. [12] proposed embedding a differentiable convex optimization layer [5] into the policy network (as the last layer) to ensure that the action complies with the constraints that form a convex set. Another approach uses the activation function in the last layer to constrain the action to a predefined range, then the action is projected to a sample-specific feasible set using a gauge function [10]. In our work, we also use an activation function to limit the range of the output. But we use linear interpolation to deduce the action value between aggregate bounds. We prefer this approach because it has lower computational overhead than differentiable layers and works well when there are only simple range constraints for the action.

2.3 Monetary Incentives for V2G

There are various approaches for incentivizing private EV owners to participate in V2G. For instance, a mechanism is designed in [43] where an aggregator dynamically changes the price to incentivize EVs to provide frequency regulation service. In [46], a two-level reverse auction is used to achieve demand response management in V2G systems. We incentivize V2G using *contract theory* [35], which is concerned with designing a set of contracts between two self-interested parties. These contracts represent a commitment where one party agrees to pay the other upon successful fulfillment of the terms of the contract. The study conducted in [21] surveys EV owners about their preferences regarding their participation in V2G contracts. It inquires about parameters such as monthly payoff, required connected time, guaranteed minimum battery level and number of discharge cycles. Jember et. al. [23] propose a two-tiered approach where the aggregator finds the optimal energy price through a game theoretic approach, and subsequently provides V2G contracts to motivate EV owners to participate. But the aggregator's interactions with the electricity markets are not studied in that work. To design V2G contracts without any prior knowledge of the EV owners' willingness to participate, Gao et. al. [18] develop an algorithm that learns the optimal unit price based on its ongoing interactions with EVs. We base our formulation of contracts on [32], where the authors design fixed-term contracts that specify the maximum discharge energy that is allowed and the corresponding payoff. In Section 3, we extend this formulation to design variable-term contracts. This requires considering two dimensions for the EV owner type. Multidimensional contracts have been studied in diverse areas, such as mobile networks [44], federated learning [27], and radio communications [11].

3 VARIABLE-TERM V2G CONTRACTS

We adopt the principal-agent model from contract theory [9] to design V2G contracts. According to this model, the VPP (principal) offers a set of V2G contracts to EV owners (agents) who decide to accept one of them or opt out based on their private information that determines their *type*.

Two-dimensional types. We assume that the agent type is two-dimensional. The first dimension, called *energy type*, indicates how much they are willing to discharge their vehicle battery. This mainly depends on how they perceive the battery degradation cost which could be influenced by various factors, from the characteristics of their vehicle battery and the difficulty of replacing it to the climate in which they live. We assume the energy type can take a finite number of values that belong to $\Theta^w = \{\theta_1^w, \ldots, \theta_I^w\}$. Note that the types are listed in ascending order, e.g. $\theta_1^w < \theta_2^w$. The second dimension, called *persistence type*, indicates how long they are willing to allow their vehicle battery to be discharged. This depends on how they perceive the cost of staying longer or idling at the charging station. We assume the persistence type can take a finite number of values that belong to $\Theta^\ell = \{\theta_1^\ell, \ldots, \theta_J^\ell\}$. The VPP operator does not know the type of a specific EV since it depends on their private information. This condition is known as *information asymmetry*. However, it knows the probability distribution over the two-dimensional types, i.e. it knows that an arbitrary EV may be of type $(\theta_i^w, \theta_j^\ell)$ with probability $\rho_{i,j}$; $(\sum_{i,j} \rho_{i,j}{=}1)$.

Contract definition. A variable-term V2G contract is characterized by a 3-tuple $(g_{i,j}, w_i, \ell_j)$, indicating respectively the payoff to agent (in €), the maximum amount of energy that can be discharged from their vehicle battery (in kWh), and the contract duration (in hours). If this contract is accepted, the VPP can withdraw up to w_i kWh from the vehicle battery during the first ℓ_j hours after accepting the contract. We seek contracts that possess two properties: *individual rationality* (IR) and *incentive compatibility* (IC). Individual rationality means that an EV owner will only accept a contract if it provides non-negative utility, that is, the payoff $g_{i,j}$ is enough to outweigh the perceived battery degradation and idling costs. Incentive compatibility guarantees that an EV gains the highest utility by choosing the contract that was specifically designed for its type. This way the contracts will be *self-revealing* [35], addressing the information asymmetry between the principal and agents.

Agent's Utility. The utility of an EV owner that accepts V2G contract $(g_{i,j}, w_i, \ell_j)$ is defined as:

$$U_{EV} = g_{i,j} - \frac{c_1 \cdot w_i}{\theta_i^w} - \frac{c_2 \cdot \ell_j}{\theta_j^\ell} \tag{1}$$

where $g_{i,j}$ is the payoff for accepting the contract and the next two terms are the cost incurred by discharging w_i from the vehicle battery over a set duration of ℓ_j. The coefficient c_1 represents the *actual* battery degradation cost measured in €/kWh. This is multiplied by the amount of discharged energy w_i and gets divided by θ_i^w, implying that this cost will be higher for lower energy types. Similarly, c_2 represents the cost of having the vehicle battery available for discharge for the first ℓ_j hours of the charging session, measured in €/hr. This is multiplied by the contract duration ℓ_j and gets divided by θ_j^ℓ, implying that this cost will be higher for lower persistence types. As θ_i^w and θ_j^ℓ appear in the denominator, higher values indicate that the EV owner is more willing to participate.

Principal's Utility. The utility function for the VPP is defined as:

$$U_{VPP} = \sum_{i=1}^{I} \sum_{j=1}^{J} \rho_{i,j} (\kappa_1 \log(w_i + 1) + \kappa_2 \log(\ell_j + 1) - g_{i,j}). \tag{2}$$

This is the VPP's expected utility, as it shows the sum of utilities over the EV types multiplied by their probability. For each EV type, the VPP appraises the amount of energy withdrawn from the battery (w_i) as well as the time window during which it can be used (ℓ_j). As the VPP is *risk-averse*, these two values are inside the concave log function. The relative importance of these two terms can be adjusted by tuning hyper-parameters κ_1 and κ_2. Finally, the payoff $g_{i,j}$ for each type must be subtracted as this corresponds to the money that will be transferred to the EV owner.

Optimal contract mechanism. The variable-term V2G contracts that will be offered by the VPP are the solution to an optimization problem that maximizes its expected utility. We overload the notation for sets Θ^w and Θ^ℓ to also denote the corresponding index sets, so we can write $i \in \Theta^w$, and $j \in \Theta^\ell$.

$$\underset{\substack{\{(g_{i,j},w_i,\ell_j)\}\\ i=1,\dots,I;\ j=1,\dots,J}}{\text{maximize}} \quad U_{VPP} \tag{3}$$

subject to:

$$\textbf{(IR)} \qquad g_{i,j} - \frac{c_1 \cdot w_i}{\theta_i^w} - \frac{c_2 \cdot \ell_j}{\theta_j^\ell} \geq 0; \quad \forall (i,j) \in \Theta^w \times \Theta^\ell$$

$$\textbf{(IC)} \qquad g_{i,j} - \frac{c_1 \cdot w_i}{\theta_i^w} - \frac{c_2 \cdot \ell_j}{\theta_j^\ell} \geq g_{i',j'} - \frac{c_1 \cdot w_{i'}}{\theta_i^w} - \frac{c_2 \cdot \ell_{j'}}{\theta_j^\ell};$$
$$\forall i, i' \in \Theta^w,\ \forall j, j' \in \Theta^\ell;\ i \neq i' \vee j \neq j'$$

$$\textbf{(PC)} \qquad w_I \leq \alpha_d \cdot \ell_J$$

$$\textbf{(MO)} \qquad 0 \leq w_1 \leq w_2 \leq \cdots \leq w_I$$
$$0 \leq \ell_1 \leq \ell_2 \leq \cdots \leq \ell_J$$
$$0 \leq g_{i,1} \leq g_{i,2} \leq \cdots \leq g_{i,J} \quad \forall i \in \Theta^w$$
$$0 \leq g_{1,j} \leq g_{2,j} \leq \cdots \leq g_{I,j} \quad \forall j \in \Theta^\ell$$

The first constraint, **(IR)**, ensures that for every type $(i,j) \in \Theta^w \times \Theta^\ell$, the contract offers non-negative utility. The second constraint, **(IC)**, ensures that an EV owner of type $(\theta_i^w, \theta_j^\ell)$ gets lower utility from accepting contract $(g_{i',j'}, w_{i'}, \ell_{j'})$ than contract $(g_{i,j}, w_i, \ell_j)$ where $i \neq i'$ or $j \neq j'$. The third constraint, **(PC)**, reflects the physical constraint of bidirectional chargers. It ensures that the maximum discharge energy w_I is less than or equal to the charger's rated discharge power α_d (measured in kWh) multiplied by the maximum contract duration ℓ_J. Thus, the VPP can effectively discharge w_I while the contract remains active. The last group of constraints, **(MO)**, ensure monotonicity of each contract parameter.

We solve the optimization problem with κ_1=0.4, κ_2=0.6, c_1=0.01 €/kWh, and c_2=0.05 €/hr. These parameters are chosen based on medium-term predictions of battery prices [19] and an analysis of rates in the electricity market similar to [32]. For the EV owner types, we are interested in three cases for each dimension, representing individuals that have low, average, and high value functions: Θ_i^w=Θ_j^ℓ={0.75, 1, 1.25}. This results in 9 distinct types. We assume a uniform distribution of types and set $\rho_{i,j}$=1/9 for all i and j values. We obtain the variable-term contracts that are shown in Tbl. 1.

	$\ell_1 = 5$	$\ell_2 = 9$	$\ell_3 = 14$
$w_1 = 19.01$	$g_{1,1} = 0.59$	$g_{1,2} = 0.79$	$g_{1,3} = 0.99$
$w_2 = 32.33$	$g_{2,1} = 0.72$	$g_{2,2} = 0.92$	$g_{2,3} = 1.12$
$w_3 = 49.00$	$g_{3,1} = 0.85$	$g_{3,2} = 1.05$	$g_{3,3} = 1.25$

Table 1: Pivot table of the contract values. Row headers indicate the amount of discharge energy in kWh, column headers indicate the contract duration in hours. The value in each cell is the payoff in €.

4 VPP OPERATION

We use a discrete time model to optimize the VPP operation. This is motivated by the fact that the VPP participates in a real-time market that has one hour resolution. In each time step (hour), the proposed algorithm goes through a loop that starts with receiving new arrivals and offering them a subset of the variable-term V2G contracts, then scheduling their charging based on the accepted contracts, stay times, energy demands, and price forecasts. Finally, the required amount of energy will be traded in the electricity market in that hour. This process is depicted in Fig. 1. We explain each of these steps below.

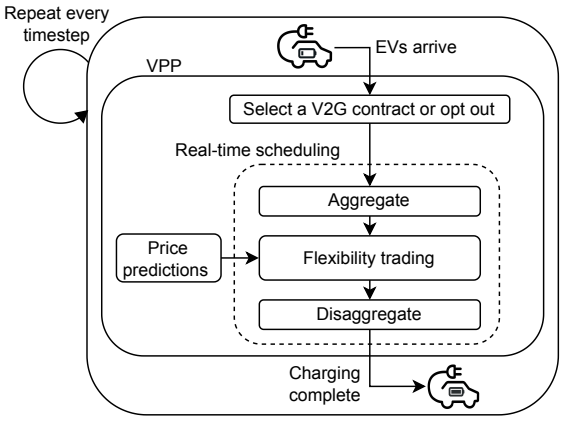

Figure 1: Overview of our methodology

4.1 Market Participation

The VPP is assumed to participate in an electricity market that runs once every hour so that electricity can be traded in real-time, and uses a single price for buy and sell in each hour. The hourly price reflects market demand and is treated as an exogenous random variable. We denote the next $h+1$ hourly prices starting from hour t in vector form as $\mathbf{p}_{t:t+h} = (p_t, \dots, p_{t+h})$. Note that every element of this vector is a random variable.

Although the trading strategy developed in this paper is suitable for participation in a single-stage electricity market, it can be used as part of a multi-stage optimization problem if the VPP participates in other electricity markets, such as the day-ahead market [32].

4.2 EV Charging Dynamics

The VPP must charge EVs in a way that ensures their energy demand will be satisfied by their specified deadline while abiding by physical limitations of charging equipment. This is referred to as a *feasible* charging schedule. Given the hourly timescale of the market, the VPP readjusts the (dis)charge power of chargers every

hour. We denote the set of EVs that are plugged in at time t as $\mathcal{N}_t$ and the set of EVs that are plugged in and have accepted a V2G contract as $\mathcal{N}_t^D \subseteq \mathcal{N}_t$. Hence, $\mathcal{N}_t \setminus \mathcal{N}_t^D$ contains all EVs that have opted out. An EV $n \in \mathcal{N}_t$ arrives in the charging station at t^n_{arr} and departs at t^n_{dep}. At any t between the arrival and departure time, its remaining connection time is given by $\tau^n_t = t^n_{dep} - t$.

The amount of energy charged or discharged from each vehicle's battery at time t forms a vector $\mathbf{y}_t = (y^1_t, \ldots, y^{|\mathcal{N}_t|}_t)$. To apply charge and discharge efficiencies, we divide each element y^n_t into a positive component (charged energy), denoted AC^n_t, and a negative component (discharged energy), denoted AD^n_t. We assume that all chargers are of the same type, with $\alpha_c > 0$ being their maximum charge power and $\alpha_d > 0$ being their maximum discharge power. Since the length of each timeslot is 1 hour, we reuse α_c and α_d to express the maximum amount of energy (in kWh) that can be charged or discharged from the battery in one timeslot. The constraints for EV charge and discharge energy are given below:

$$y^n_t = AC^n_t + AD^n_t \qquad \forall n \in \mathcal{N}_t \tag{4a}$$
$$0 \le AC^n_t \le \alpha_c \qquad \forall n \in \mathcal{N}_t \tag{4b}$$
$$AD^n_t = 0 \qquad \forall n \in \mathcal{N}_t \setminus \mathcal{N}_t^D \tag{4c}$$
$$-\alpha_d \le AD^n_t \le 0 \qquad \forall n \in \mathcal{N}_t^D \tag{4d}$$

Note that all of these constraints are defined only at times when EV n is connected to a charger, i.e. for $t \in \{t^n_{arr}, \ldots, t^n_{dep}\}$. We denote the energy capacity of its battery as B^n and its SOC at time t as soc^n_t. At t^n_{arr}, soc^n_t is initialized with the observed SOC at arrival, soc^n_{arr}. For the schedule to be feasible, by t^n_{dep}, soc^n_t must reach the SOC specified by the EV owner, denoted as soc^n_{dep}. Furthermore, soc^n_t must be maintained between minimum and maximum levels, δ_{min} and δ_{max}, at all times. While connected, soc^n_t is calculated using a recursive formula after accounting for the energy charged or discharged by the VPP and incorporating the battery charge and discharge efficiencies, denoted as η_c and η_d respectively. We assume that the battery self-discharge is negligible at this timescale and can be ignored. The SOC constraints are given below:

$$\delta_{min} \le soc^n_t \le \delta_{max} \qquad \forall n \in \mathcal{N}_t \tag{5a}$$
$$soc^n_t = soc^n_{arr} \qquad \text{when } t = t^n_{arr} \tag{5b}$$
$$soc^n_t = soc^n_{dep} \qquad \text{when } t = t^n_{dep} \tag{5c}$$
$$soc^n_{t+1} = soc^n_t + \frac{\eta_c\, AC^n_t}{B^n} + \frac{AD^n_t}{\eta_d\, B^n} \qquad \forall n \in \mathcal{N}_t \tag{5d}$$

Lastly, the accepted V2G contracts pose additional constraints on the charging schedule. Let us use $\bar{w}^n_t$ and $\bar{\ell}^n_t$ to track the remaining energy that can be discharged and the remaining time to discharge this energy from the battery of the n^{th} EV. If this EV refuses to accept a contract, i.e. $n \in \mathcal{N}_t \setminus \mathcal{N}_t^D$, then both $\bar{w}^n_t$ and $\bar{\ell}^n_t$ are set to zero. Recall that w_i and ℓ_j are the initial parameters of the V2G contract. Thus, when EV n arrives at t^n_{arr} and accepts contract $(g_{i,j}, w_i, \ell_j)$, then we initialize $\bar{w}^n_t = w_i$ and $\bar{\ell}^n_t = \ell_j$. It follows from these definitions that every time the battery of this EV is discharged, we have to update $\bar{w}^n_t$ by subtracting the amount of energy that is discharged. Moreover, $\bar{\ell}^n_t$ is reduced by one in every timeslot regardless of whether the battery is discharged. An EV can be discharged only if $\bar{w}^n_t > 0$ and $\bar{\ell}^n_t > 0$. If one of them reaches zero, the EV is removed from $\mathcal{N}_t^D$ and put into $\mathcal{N}_t \setminus \mathcal{N}_t^D$, then both $\bar{w}^n_t$ and $\bar{\ell}^n_t$ are set to zero.

$$\bar{w}^n_{t+1} = \bar{w}^n_t + AD^n_t/\eta_d \qquad \forall n \in \mathcal{N}_t^D \tag{6a}$$
$$\bar{\ell}^n_{t+1} = \bar{\ell}^n_t - 1 \qquad \forall n \in \mathcal{N}_t^D \tag{6b}$$
$$-AD^n_t \le \bar{w}^n_t \eta_d \qquad \forall n \in \mathcal{N}_t^D \tag{6c}$$

Note how the discharge efficiency η_d is taken into account in Eqn.(6c). This is because the contract specifies how much energy can be withdrawn directly from the battery. But due to discharge inefficiency, the VPP will receive slightly less energy.

In summary, for EV n at time t, the variables that are updated at every timestep are: (i) state-of-charge, soc^n_t; (ii) time left until departure, τ^n_t; (iii) remaining energy available for discharge, $\bar{w}^n_t$; (iv) remaining term of the contract, $\bar{\ell}^n_t$.

4.3 Pruning V2G Contracts

At the start of each timeslot t, the VPP receives the list of newly connected EVs along with their battery capacity B^n, initial SOC soc^n_{arr}, arrival time t^n_{arr}=t, specified departure time t^n_{dep}, and desired SOC at departure soc^n_{dep}. With this information, the VPP calculates the *laxity* of each EV which is defined as the difference between the time left until departure and the minimum amount of time that is required to bring the battery SOC to their desired SOC. Calculating laxity is helpful because it must remain nonnegative at all times for the charging schedule to be feasible [31].

$$lax^n_t = \tau^n_t - \frac{(soc^n_{dep} - soc^n_t)B^n}{\alpha_c \eta_c} \qquad \forall n \in \mathcal{N}_t \tag{7a}$$
$$lax^n_t \ge 0 \qquad \forall n \in \mathcal{N}_t \tag{7b}$$

We assume that EVs arrive with non-negative laxity, so initially there is at least one feasible charging schedule. Note that the laxity at arrival can be calculated by substituting t^n_{arr} for t in Eqn. (7a).

Once the laxity is calculated for each EV, the VPP offers a subset of V2G contracts presented in Tbl. 1 by pruning the V2G contracts that, if accepted, there will be no feasible charging schedule for the respective EV. To find this subset, the VPP performs three entry checks: (i) The EV does not depart before the contract term is over, $t^n_{dep} - t^n_{arr} \ge \ell_j$. (ii) The energy content of its battery at arrival is not less than the contract discharge energy, $B^n soc^n_{arr} \ge w_i$. (iii) The time required to discharge w_i from the battery and then charge it by the same amount does not exceed its laxity at arrival: $w_i \cdot \eta_d/\alpha_d + w_i/(\alpha_c \cdot \eta_c) \le lax^n_{t_{arr}}$.

If there are no feasible contracts or none of them provides positive utility to the EV owner, then they will opt out of V2G. In that case, their battery is still charged to the desired SOC, and their laxity is used to offer flexibility, which is less than the flexibility they could offer if they permitted bidirectional charging. Otherwise, the EV owner will choose the contract that corresponds to their type $(\theta^w_i, \theta^\ell_j) \in \Theta^w \times \Theta^\ell$ as the contracts are incentive compatible. If the contract designed specifically for their type is not feasible, the EV owner will choose one with a lower discharge energy or a shorter term (whichever maximizes their utility), as long as it is feasible and provides positive utility as outlined in Eqn. (1).

4.4 Optimal VPP Scheduling

Once the set of newly arrived EVs that accepted a V2G contract is determined, the VPP decides how much energy it should trade in in the real-time market by solving an optimization problem to maximize its total profit by minimizing the cost of trading in the market. Indeed, this problem is a stochastic optimization problem due to the uncertainty of EV mobility and hourly prices. A fast and robust approach to trading flexibility under uncertainty will be discussed in the next section. For now, we assume perfect information about the future and solve this problem in an offline fashion. This serves as a baseline enabling us to quantify the optimality gap due to uncertainty in Section 6.

The cost of trading in the market depends on the net energy delivered to EVs in that timeslot:

$$y_t^{agg} = \sum_{n \in \mathcal{N}_t} y_t^n \tag{8}$$

Note that y_t^{agg} would be negative if the amount of energy discharged from EVs surpassed the amount of energy used to charge them.

Problem (9) finds the charging schedule that minimizes the cost of trading in the market from the current time $t_\emptyset$ up to the next h hours. In order for the solution to be the true optimal, the optimization horizon should be the maximum departure time of all the cars currently connected. That is $h = \max(\tau_t^n,\ \forall n \in \mathcal{N}_t)$. The cost is calculated from multiplying the predicted market price for that hour by the net energy delivered to EVs in that hour, y_t^{agg}. For the solution to be the optimal, price predictions are assumed to be perfect. The main optimization variable is a matrix containing the charging amount for every currently connected EV $n \in \mathcal{N}_{t_\emptyset}$ from the current time $t_\emptyset$ up to $t_\emptyset + h$, that is $\mathbf{Y}_{t_\emptyset:t_\emptyset+h} = \left[\mathbf{y}_{t_\emptyset}, \ldots, \mathbf{y}_{t_\emptyset+h}\right]$, with each $\mathbf{y}_t = \left[y_t^1, \ldots, y_t^{|\mathcal{N}_{t_\emptyset}|}\right]$. Note that $\mathbf{AC}_{t_\emptyset:t_\emptyset+h}$ and $\mathbf{AD}_{t_\emptyset:t_\emptyset+h}$ are auxiliary variables that determine $\mathbf{Y}_{t_\emptyset:t_\emptyset+h}$.

$$\underset{\mathbf{Y}_{t_\emptyset:t_\emptyset+h};\, \mathbf{AC}_{t_\emptyset:t_\emptyset+h};\, \mathbf{AD}_{t_\emptyset:t_\emptyset+h}}{\text{minimize}} \quad \sum_{t=t_\emptyset}^{t_\emptyset+h} \hat{p}_t \cdot y_t^{agg} \tag{9}$$

subject to:

Charging constraints Eqns. (4)

State-of-charge constraints Eqns. (5)

V2G contract constraints Eqns. (6)

Remark 1. In the above problem, we do not need a constraint to explicitly prevent a battery from being charged and discharged at the same time, i.e. forcing $AC_t^n AD_t^n = 0$. Due to battery imperfections, such opposing actions would waste energy and are therefore suboptimal.

Remark 2. Since there are no coupling constraints between EVs in Problem (9), the VPP does not need predictions for arrivals that will occur between $t_\emptyset$ and $t_\emptyset + h$ in order to find the optimal solution for the current time $t_\emptyset$.

Observe that in Problem (9), the objective function is linear and all constraints are affine. Thus, it is a linear program that can be solved in polynomial time using interior point or simplex-based methods. Nevertheless, when the VPP aggregates a large number of EVs and some of them remain connected to a charger for an extended period of time, solving this problem with noisy or perfect predictions takes a considerable amount of time because of the optimization horizon and the number of decision variables and constraints.

Let us denote the solution of the above optimization problem as $\mathbf{Y}^\star_{t_\emptyset:t_\emptyset+h}$. The first row of this matrix, $\mathbf{y}^\star_{t_\emptyset}$, denotes the optimal decision that must be implemented at $t = t_\emptyset$.

5 TRADING FLEXIBILITY IN AGGREGATE

We present our methodology for solving the stochastic optimization problem presented in Section 4. Our goal is to design an algorithm for flexibility trading that does not rely on accurate forecasts, and its running time does not change drastically as more EVs are controlled by the VPP. Moreover, it should guarantee that there is a feasible charging schedule for all EVs regardless of whether they accepted a V2G contract, and that EVs are treated fairly. To achieve this goal, we attempt to trade flexibility in aggregate using an RL agent that, once trained, can be quickly executed.

The proposed methodology has three steps which are shown inside the dashed box in Fig. 1. First, we create an aggregate representation of the state, denoted as $\mathbf{s}_t^{agg}$, by aggregating individual EVs into a virtual battery. In this process, we also aggregate constraints defined for individual EVs to ensure that the action taken for the virtual battery will not violate a constraint defined for an EV. In the second step, we pass the aggregate state representation to a safe RL agent that outputs an action to operate the virtual battery. This action is denoted as y_t^{agg} and minimizes the expected cost of trading in the market. Due to the way that we aggregate constraints and the design of our RL agent, this action is guaranteed to satisfy constraints defined for individual EVs after it is disaggregated to their respective actions. In the last step, we disaggregate y_t^{agg} into $(y_t^1, \ldots, y_t^{|\mathcal{N}_t|})$ by incorporating a fairness criterion.

5.1 Aggregation

Aggregating the energy storage capacity of individuals EVs into a virtual battery is essential for using a single-agent reinforcement learning framework for real-time scheduling of flexibility, independent of the number of connected EVs. The main challenge in learning an action for (dis)charging this virtual battery is ensuring that it does not violate the feasibility requirement of individual charging schedules. In this section, we show that this can be achieved by carefully aggregating the constraints defined for the (dis)charge action of each individual EV, i.e. y_t^n, in Problem (9). Specifically, we inspect the constraints defined for individual EVs and identify those that can be translated to upper and lower bounds for y_t^n. This allows us to aggregate lower and upper bounds separately to obtain bounds for y_t^{agg}.

We first examine Eqn. (4b) which states that $0 \le AC_t^n \le \alpha_c$. Since we know from Eqn. (4a) that $y_t^n = AC_t^n + AD_t^n$ and that AC_t^n and AD_t^n cannot be nonzero at the same time at the optimal point (see Remark 1), we can rewrite (4b) as $0 \le y_t^n \le \alpha_c$ when the battery is charging. From this we get an upper bound on y_t^n:

$$y_{t,upper1}^n = \alpha_c \quad \forall n \in \mathcal{N}_t \tag{10}$$

Using the same arguments, we get a lower bound on y_t^n from combining Eqns. (4c) and (4d):

$$y_{t,lower1}^n = -\alpha_d \quad n \in \mathcal{N}_t^D \tag{11}$$

For EVs that accepted a V2G contract, Eqn. (6c) forces the VPP to respect the discharge energy in the contract. This gives the following lower bound:

$$y^n_{t,lower2} = -\bar{w}^n_t \eta_d \quad n \in \mathcal{N}^D_t \tag{12}$$

Next, we consider the SOC constraints in (5a) and (5d). By substituting (5d) in (5a) for the two cases where the battery charges and discharges, we get an upper bound and a lower bound on y^n_t:

$$y^n_{t,upper2} = B^n(\delta_{max} - soc^n_t)/\eta_c \qquad \forall n \in \mathcal{N}_t \tag{13}$$

$$y^n_{t,lower3} = B^n \eta_d(\delta_{min} - soc^n_t) \qquad \forall n \in \mathcal{N}^D_t \tag{14}$$

Finally, we turn our attention to constraints (7a) and (7b) which deal with laxity. Notice that y^n_t should maintain the non-negativity of laxity in the next step, as $lax^n_{t+1} \geq 0$ indicates that it is possible for the SOC of this EV to reach soc^n_{dep} by t^n_{dep}. To get the definition of lax^n_{t+1}, we write (7a) for $t+1$. We then substitute (5d) in (7a) to relate laxity at $t+1$ to the action taken at t, and replace τ^n_{t+1} with $\tau^n_t - 1$. To obtain a lower bound for y^n_t, we set lax^n_{t+1} to zero. Finally we consider the two cases where $y^n_t = AC^n_t > 0$ and $y^n_t = AD^n_t < 0$ separately as the efficiencies are handled differently in Eqn. (5d). Solving each case for y^n_t, we get Eqns. (14) and (15):

$$y^n_{t,lower4} = B^n(soc^n_{dep} - soc^n_t)/\eta_c - \alpha_c(\tau^n_t - 1) \qquad \forall n \in \mathcal{N}_t \tag{15}$$

$$y^n_{t,lower5} = B^n \eta_d(soc^n_{dep} - soc^n_t) - \alpha_c \eta_c \eta_d(\tau^n_t - 1) \quad \forall n \in \mathcal{N}^D_t \tag{16}$$

Now that we have all constraints posed on y^n_t, we can write the overall individual upper and lower bounds as follows:

$$\begin{aligned}
y^n_{t,upper} &= \min\left(y^n_{t,upper1}, y^n_{t,upper2}\right) && \forall n \in \mathcal{N}_t \\
y^n_{t,lower} &= y^n_{t,lower4} && \forall n \in \mathcal{N}_t \setminus \mathcal{N}^D_t \\
y^n_{t,lower} &= \max\left(y^n_{t,lower1}, \ldots, y^n_{t,lower5}\right) && \forall n \in \mathcal{N}^D_t
\end{aligned}$$

Finally, we get the upper (lower) bound for action of the virtual battery, y^{agg}_t, by aggregating the upper (lower) bounds of individual EVs:

$$\sum_n y^n_{t,lower} \leq y^{agg}_t \leq \sum_n y^n_{t,upper} \tag{17}$$

It follows from the above derivations that if the y^{agg}_t returned by the RL policy satisfies (17), there exists a disaggregation of y^{agg}_t into $(y^1_t, \ldots, y^{|\mathcal{N}_t|}_t)$ such that every element of this vector satisfies the feasibility requirement of the respective EV charging schedule. We denote the aggregate lower and upper bounds in Eqn. (17) as $y^{agg}_{t,lower}$ and $y^{agg}_{t,upper}$, respectively.

5.2 Soft Actor-Critic Reinforcement Learning

To schedule charge and discharge of the virtual battery, we use the Soft Actor-Critic (SAC) algorithm which trains a stochastic policy with entropy regularization [20]. Our SAC agent, called *Aggregate SAC*, receives the aggregate state representation (defined below) and outputs the (dis)charge action of the virtual battery. This action determines the amount of energy that will be traded in the market in real-time. The three main components of Aggregate SAC are the actor, critic, and replay buffer, as shown in Fig. 2.

5.2.1 Aggregate State. To make the state of the SAC agent truly independent of the number of connected EVs, we argue that the number of state variable (dimension of the state space) and the scale of each state variable should be independent of the number of connected EVs. This leads us to design an aggregate *state* representation denoted as $\mathbf{s}^{agg}$. Specifically, we take the average of features characterizing individual EVs, such as SOC, upper and lower bounds of the action, etc. Additionally, price predictions for the current time step and the next h time steps, $\hat{\mathbf{p}}_{t:t+h}$, and its discrete derivative, $\Delta\hat{\mathbf{p}}_{t:t+h}/\Delta t$, are included in the state. Please refer to Appendix A for the full list of state variables. All of these variables are concatenated and the resulting 1D vector is fed to the agent.

5.2.2 Reward. The reward signal is defined as $r_t = -p_{t-1} \cdot y^{agg}_{t-1}$ which is the negated single-step version of the objective function of Problem (9). As the SAC agent maximizes the cumulative reward over an episode, the reward must be defined as the negative transfer to the imbalance market at a single timeslot t. Note that $r_t > 0$ implies that the VPP receives money from trading in the market, while $r_t < 0$ implies that the VPP pays for the trade.

5.2.3 Actor and Critic Networks. The actor is a neural network that takes the aggregate state $\mathbf{s}^{agg}_t$, and producing the mean and standard deviation, μ_t and σ_t, of a Gaussian distribution from which a coefficient $\tilde{\beta}_t$ is sampled. We use the *sigmoid* activation function in the last layer before outputting μ_t to ensure that this coefficient falls in the range of $[0, 1]$. The sampled $\tilde{\beta}_t$ is then clipped between 0 and 1 to obtain the final coefficient β_t. Subsequently, β_t is used for linear interpolation between the upper and lower bounds, $y^{agg}_{t,upper}$ and $y^{agg}_{t,lower}$ respectively:

$$y^{agg}_t = \beta_t \cdot y^{agg}_{t,upper} + (1 - \beta_t) \cdot y^{agg}_{t,lower} \tag{18}$$

The resulting y^{agg}_t represents the total amount of energy that must be charged or discharged from the virtual battery at t. This value is then passed to the disaggregation algorithm described in Section 5.3) to determine individual (dis)charging actions.

The other two components, the critic and replay buffer, play central roles in the model-free RL algorithm outlined in [20]. Essentially, the the SAC algorithm seeks to find the policy that maximizes an objective function, combining the accumulated expected reward and the entropy of the policy at any given state. The entropy encourages the policy to act more randomly, promoting exploration. The extent to which the entropy is considered is tuned by a temperature term. The critic learns a value function, indicating the expected reward of selecting an action at a given state. This helps to provide learning updates to the actor to enhance the policy. In our implementation, tuples of state, action, reward, and next state are saved to the buffer, and subsequently sampled to perform updates to the actor and the critic (labeled in Fig. 2 as $(\mathbf{s}^{agg}_i, \tilde{\beta}_i, r_i, \mathbf{s}^{agg}_{i+1})$).

5.3 Proportionally Fair Disaggregation

The VPP employs a disaggregation algorithm to compute $\mathbf{y}_t = (y^1_t, \ldots, y^{|\mathcal{N}_t|}_t)$, which is a feasible charging schedule for individual EVs, given y^{agg}_t returned by the RL agent. While there are various ways to disaggregate y^{agg}_t, we compute $\mathbf{y}_t$ by solving a convex optimization problem. In particular, we maximize the sum of the

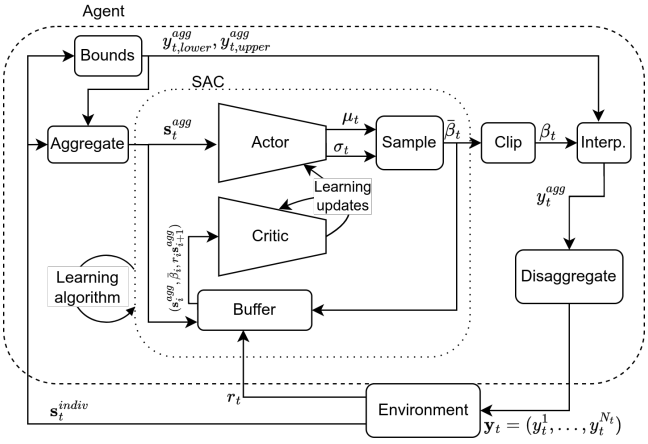

Figure 2: Block diagram of the agent, *Aggregate SAC*.

logarithm of every EV owner's utility function, where the utility function of EV owner n is given by:

$$U^n_t(y^n_t) = y^n_t - y^n_{t,lower} + 1 \tag{19}$$

This utility function is greater than or equal to 1 and increasing in y^n_t. We wish to emphasize that this function is different from (1) and is defined for all EVs regardless of whether they have accepted a V2G contract. The following optimization problem is solved at every timeslot.

$$\underset{\mathrm{y}_t=\left(y^1_t,\ldots,y^{|\mathcal{N}_t|}_t\right)}{\text{maximize}} \quad \sum_{n\in\mathcal{N}_t} \log\left(U^n_t(y^n_t)\right) \tag{20a}$$

subject to:

$$\sum_{n\in\mathcal{N}_t} y^n_t = y^{agg}_t \tag{20b}$$

$$y^n_{t,lower} \le y^n_t \le y^n_{t,upper} \qquad \forall n \in \mathcal{N}_t \tag{20c}$$

The significance of this formulation is that the solution of this convex problem, which we call a *proportionally fair* disaggregation, satisfies fairness axioms from game theory [45]. We have borrowed the notion of proportional fairness from the resource allocation literature to ensure that all EVs are treated fairly.

5.3.1 Priority Based Disaggregation. To evaluate the performance of the *proportionally fair* disaggregation algorithm, we borrow a priority-based resource allocation algorithm and use it for flexibility disaggregation. The basic idea of a priority-based algorithm is that we can assign an arbitrary scalar that signifies priority to each EV n at every timeslot t. Let us denote this priority as pr^n_t. The algorithm sorts the EVs from highest to lowest priority and they receive their share of y^{agg}_t in that order. Concretely, the algorithm receives the priority list $\mathbf{pr}_t$, the aggregate action of the virtual battery (y^{agg}_t) which is either charge or discharge this amount of energy depending on its sign, and the upper and lower bounds of the action of each EV (y^n_t). Next, it starts by fulfilling the lower bound of each EV. If there is surplus energy after that, the maximum possible energy is allocated to the EV with the highest priority. It continues down the list until y^{agg}_t is used up. The pseudocode for this algorithm is presented in Appendix B. In Section 6, we define the priority of each EV based on its laxity. This results in two versions of this algorithm, namely *most laxity first* and *least laxity first*.

6 EXPERIMENTS

6.1 Datasets & Baselines

In this section we evaluate the proposed methods for energy trading in the real-time market through aggregation and disaggregation of flexibility provided by individual EVs. We use two real datasets that contain data collected between Jan. 1 and Dec. 31, 2019 from the Netherlands. The first dataset contains settlement prices in the imbalance market operated by TENNET [3]. This market has a 15 minute timescale, so to make it compatible with our 1-hour timeslots, we take the price at the top of every hour. This results in a total of $365 \times 24 = 8,760$ data points.

The second dataset is the ElaadNL dataset [2]. It contains 10,000 charging sessions that took place in 2019 in a network of public charging stations in the Rotterdam region. Each session is characterized by a charger ID in addition to the arrival time t^n_{arr}, departure time t^n_{dep}, and amount of energy delivered to the EV that connected to this charger. To calculate the SOC at arrival, we assume each EV was charged to $soc^n_{dep} = 0.97$ before departure. Then, we calculate its soc^n_{arr} based on the amount of energy delivered in that charging session. We assume all chargers support bidirectional charging, their rated charge and discharge power is $\alpha_c, \alpha_d = 11$ kW, and the battery charge and discharge efficiency is $\eta_c, \eta_d = 0.98$. The SOC bounds are set to $\delta_{min} = 0$ and $\delta_{max} = 1$, and the energy capacity of the battery is set to $B = 80$ kWh.[2] While the methodology is designed to handle a heterogeneous set of EVs, we chose this to simplify the data pre-processing. As discussed in Section 4, we assume that all EVs have non-negative laxity upon arrival, so we discard all EVs with negative laxity in the dataset. That leaves us with 9,997 sessions. Plots depicting the main characteristics of the prices and charging sessions can be found in Appendix C.

When the EVs arrive to charging stations operated by the VPP, their type with respect to energy and persistence is sampled from a discrete uniform distribution between 1 and 3. We build the contract menu for each EV based on their specified charging deadline and energy demand after applying the entry checks outlined in Section 4.3. As a result, a subset of the nine contracts presented in Tbl. 1 will be offered to that EV.

As we need some of the data to train the RL agent, we split the price and EV charging datasets into training and test. The training dataset spans from January to June, while the test dataset spans from July to December. Thus, the training set contains 4,764 sessions, and the test set contains 5,233 sessions. Unless otherwise stated, the main disaggregation algorithm that is used in our experiments is the *proportionally fair* algorithm, and the real-time scheduling algorithm that is based on this disaggregation scheme is called *Aggregate SAC PF*.

Baselines. We use three baselines to evaluate our approach. The first is *No-Control*, when an EV is charged at the maximum power supported by the charger as soon as it connects to the charger.

[2]This is approximately the capacity of common EV models such as Hyundai IONIQ 5 (77.4 kWh), Tesla Model 3 Long Range (82 kWh), and Ford Mustang Mach E (88 kWh).

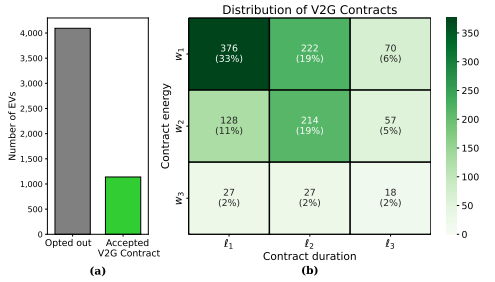

Figure 3: Plot (a) shows EVs that opted out or accepted a V2G contract; (b) shows the distribution of accepted contracts.

This 'charging as soon as possible' policy minimizes the length of the charging session without taking advantage of any kind of flexibility. The second is *No-V2G* which solves Problem (9) with perfect information, but without offering any V2G contracts. In this case, the VPP only uses the flexibility allowed by the EVs' laxity at arrival. Lastly, we have *OPT-V2G* which solves Problem (9) in an offline fashion, taking into account the V2G contracts. As we solve this problem with perfect information, it gives the best solution which can be possibly achieved. For a fair comparison with *Aggregate SAC PF*, we also solve Problem (9) with noisy price predictions. In that case, we renamed it to *LP-V2G* since it no longer represents the optimal solution.

Implementation details. The VPP is implemented in an in-house environment developed in Python 3.10 on an Ubuntu 22 machine with an AMD EPYC 7313 CPU (64 cores), on-board ASPEED GPU, and 516 GB of RAM. We model the optimization problems (3), (9) and (20) in CVXPY [17] and solve them using Mosek [8]. The implementation of *Aggregate SAC* agent is based on the code provided by CleanRL's continuous-action SAC [22] for PyTorch [29].

6.2 Contract Acceptance

We first analyze how the EV owners accept or opt out of the variable-term V2G contracts. Fig. 3 depicts the distributions of V2G contracts that were accepted in our experiment. Interestingly, most EV owners opted out and only 21% of them accepted a V2G contract. This is because their energy or time type creates a contract menu without any feasible contracts or with contracts that provide negative utility. For example, many EVs do not have enough laxity at arrival to participate in V2G for 5 hours or more. This is evident from Fig. 7c. Focusing on the EV owners that accepted a contract, we see that the most popular contract is $(g_{1,1}, w_1, \ell_1) = (0.59\ €, 19.01\ \text{kWh}, 5\ \text{hr})$ with 33% of the total share of accepted contracts. As the contract increases in energy or duration, it is selected less frequently.

6.3 Evaluation of Aggregate Trading Agent

Training and testing. The *Aggregate SAC PF* agent is trained via interaction with the environment in discrete time steps, where each episode of interaction starts on January 1, 2019 and ends on June 30, 2019. We consider 200 episodes for training and deploy the policy obtained at the end of episode 200 to trade EV charging flexibility in the imbalance market during the second half of the year (July to December 2019). Thus, the training and testing episodes will have nearly the same number of 1-hour time steps, that is 4,344 and 4,416 time steps respectively. We continue making learning updates after the agent is deployed, provided they are done based on information we have seen from deployment till now.

Fig. 4a shows the learning curve of the Aggregate SAC PF agent. The agent is trained on perfect price predictions. In accordance with best practices [30], we perform five independent runs. The shaded area shows the cost (lower is better) of the best and worst performing agents in every episode (i.e. the tolerance bounds), and the solid line shows the average performance. The horizontal dotted lines show the baseline performance for comparison. We can see that there is some variations in the agents during the early episodes (5-25), but the gap narrows as more episodes are used for training. Furthermore, this figure shows that during training, the *Aggregate SAC PF* settles on a performance around 40% higher than *OPT-V2G*.

When we evaluate the agent, we consider different qualities of price forecast. These are generated by adding synthetic noise to the prices: $\hat{p}_t = p_t + \epsilon_t$. Here, ϵ_t is sampled from a normal distribution with mean of 0 and a standard deviation $\sigma = 0, 0.01, 0.02, 0.04, 0.06$. The case where $\sigma = 0$ corresponds to perfect predictions, and the other ones correspond to progressively worse predictions.

After letting the agent train for 200 episodes (using perfect information during training), we perform five runs on each of our forecast scenarios. These are shown on Fig. 4b, where the x-axis shows the amount of noise added to the predictions when the agent is deployed, and the y-axis shows the transfer to the imbalance market during the test episode. A small offset along the x-axis is added to the points to avoid overlaps. As before, the shaded area shows the best and worst performance. We compare the performance of *Aggregate SAC PF* to the two baselines, which were run five times as well. Recall that *LP-V2G* solves the same problem as *OPT-V2G*, but with noisy predictions.

When predictions are perfect ($\sigma = 0$), *Aggregate SAC PF* achieves a transfer to market that is lower (better) than *No-V2G* and, as expected, higher than *LP-V2G* which is equivalent to *OPT-V2G* in this case. The good performance of *Aggregate SAC PF* shows that it is able to translate what it learned during training to the new charging sessions and prices in the test set. Therefore, it is robust to distribution shifts.[3] Moreover, it can be seen that as the prediction quality decreases, the gap between *Aggregate SAC PF* and *LP-V2G* quickly narrows. When the noise is very high, $\sigma = 0.04, 0.06$, *Aggregate SAC PF* achieves marginally better performance than *LP-V2G*. This shows that *Aggregate SAC PF* is more robust to imperfect price forecasts than the baselines.

6.4 Profitability

The total VPP's profit is calculated by subtracting the transfer to the imbalance market and payoffs for the accepted V2G contracts from the revenue received from the EVs for supplying their charging demand. We assume EV owners pay a fixed retail price, p^R, for the energy delivered to them, which is $B^n(soc^n_{dep} - soc^n_{arr})$. The rate is $p^R = 0.064$ €/kWh which corresponds to the 85^{th} percentile of the imbalance price over the year.

In Fig. 5, we compare the profit made by our top performing *Aggregate SAC PF* on the test dataset, under imperfect price forecasts

[3] Note the difference in prices from Jan.-Jun. to Jul.-Dec. in Fig. 6, and the difference in daily charging sessions between the first and second halves of the year in Fig. 7a.

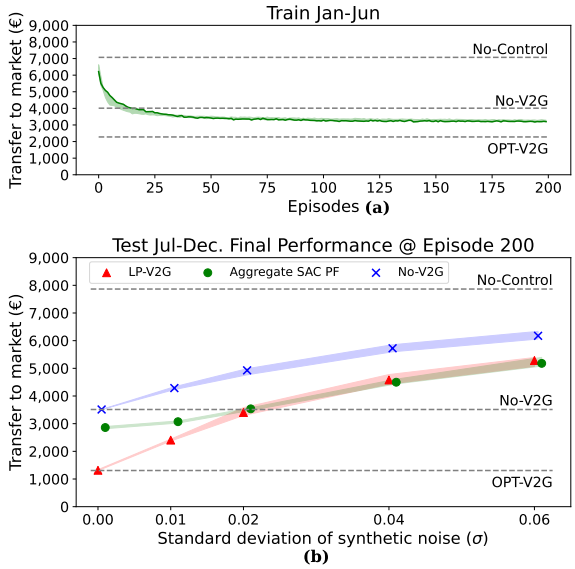

Figure 4: *Aggregate SAC PF* with perfect and noisy predictions. Subplot (a) shows the learning curve during training. Subplot (b) shows the final test performance compared with baselines under different forecasting scenarios.

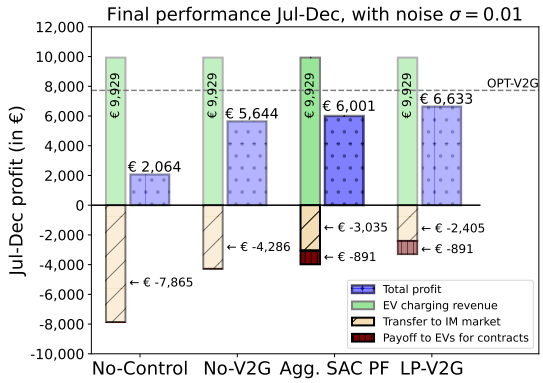

Figure 5: Performance for the baselines and top performing *Aggregate SAC PF* on the test dataset (Jul-Dec) with noise level σ=0.01.

with the smallest noise, i.e. σ=0.01. The *No-Control* baseline has the lowest profit due to its inability to control EV charging. Next is the *No-V2G* baseline that achieves a higher profit by reducing the transfer to the market to approximately 45% of the previous baseline, underscoring the significance of unidirectional flexibility. Further reduction in the transfer to market is achieved by *Aggregate SAC PF*, precisely 29% compared to *No-V2G*, resulting in higher profitability. This suggests that, despite the payoffs given to EVs, V2G contracts lead to increased profitability. By using information about individual EVs and knowledge of the state evolution model, *LP-V2G* attains the highest profit. Again, the performance of *Aggregate SAC PF* is noteworthy, given its model-free approach and the fact that it utilizes aggregate state information instead of information about every individual EV and considers only 1-step ahead flexibility. Finally, the profitability of *OPT-V2G* is shown as a horizontal dotted line.

Disaggregation Algorithm	Min.	Mean	Max.
Least Laxity First	3,043	**3,066**	**3,118**
Proportionally Fair	**3,035**	3,086	3,152
Most Laxity First	3,266	3,357	3,459

Table 2: One-shot performance (transfer to market in €) of different disaggregation algorithms in the imbalance market with noise level σ=0.01.

6.5 Performance of Disaggregation Algorithms

To evaluate the performance of the *Proportionally Fair* disaggregation algorithm, we utilize the five *Aggregate SAC PF* agents that we previously trained. The evaluation on the test dataset with imperfect price predictions, σ=0.01, is repeated once more, but this time, disaggregation is done using the *priority based* algorithms described in Section 5.3.1.

The results are presented in Tbl. 2. Among these, the *Least Laxity First* is the most effective one in terms of the average performance, showcasing a slight improvement over *Proportionally Fair*. On the other hand, *Most Laxity First* performs the least favorably, accumulating around 8.7% higher cost than *Proportionally Fair*. These findings showcase how the choice of the disaggregation algorithm affects performance, as it decides which specific EVs receive energy, thereby changing the flexibility available in future time steps to the RL agent. Nonetheless, it can be argued that the fairness property offered by *Proportionally Fair* disaggregation is advantageous, considering its performance is on a par with *Least Laxity First*, which does not take fairness into account.

7 CONCLUSION

We proposed a real-time scheduling algorithm for a VPP that trades aggregate flexibility in a real-time electricity market with hourly resolution. The VPP procures V2G participation by offering contracts to EV owners. This increases its profitability despite offering V2G incentives. At the core of our scheduling algorithm is a model-free RL agent capable of making decisions on behalf of all EVs that are controlled by the VPP and trading their flexibility. This is accomplished by aggregating them into a virtual battery with specific constraints. This aggregation is performed in a way that ensures the aggregate action results in a feasible schedule for all EVs. The aggregate action is then broken into individual actions using a proportionally fair disaggregation scheme. Our evaluation shows remarkable performance of this real-time scheduling algorithm, especially when it receives noisy forecasts for future hours.

For future work, we intend to conduct tests on datasets featuring a higher number of charging sessions per day (10x-100x) to explore the scalability of the proposed algorithm and measure any potential performance degradation. Additionally, we plan to compare our result with the solution found via robust optimization or risk-aware reinforcement learning as they are expected to better handle uncertainties arising from market prices and EV traffic patterns.

REFERENCES

[1] [n. d.]. Bidirectional EV chargers to finally materialize in 2024. https://www.solarpowerworldonline.com/2024/01/bidirectional-ev-chargers-to-finally-materialize-in-2024/

[2] [n. d.]. ElaadNL Open Data. https://platform.elaad.io/download-data/

[3] [n. d.]. Tennet Export data. https://www.tennet.org/english/operational_management/export_data.aspx

[4] [n. d.]. U.S. electricity customers averaged five and one-half hours of power interruptions in 2022. https://www.eia.gov/todayinenergy/detail.php?id=61303

[5] Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J Zico Kolter. 2019. Differentiable convex optimization layers. *Advances in neural information processing systems* 32 (2019).

[6] Abdullah Al Zishan, Moosa Moghimi Haji, and Omid Ardakanian. 2020. Adaptive Control of Plug-in Electric Vehicle Charging with Reinforcement Learning. In *Proceedings of the Eleventh ACM International Conference on Future Energy Systems* (Virtual Event, Australia) *(e-Energy '20)*. ACM, 116–120.

[7] Bushra Alshehhi, Areg Karapetyan, Khaled Elbassioni, Sid Chi-Kin Chau, and Majid Khonji. 2023. DClEVerNet: Deep Combinatorial Learning for Efficient EV Charging Scheduling in Large-scale Networked Facilities. In *Proceedings of the 14th ACM International Conference on Future Energy Systems*. 287–298.

[8] MOSEK ApS. 2022. Mosek optimizer API for Python. *Version* 9, 17 (2022), 6–4.

[9] Patrick Bolton and Mathias Dewatripont. 2004. *Contract theory*. MIT press.

[10] Mathis Brosowsky, Florian Keck, Olaf Dünkel, and Marius Zöllner. 2021. Sample-specific output constraints for neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 6812–6821.

[11] Yanming Cao, Qi Shi, Xinbing Wang, Xiaohua Tian, and Yu Cheng. 2012. Two-dimensional contract theory in cognitive radio networks. In *2012 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 1156–1161.

[12] Bingqing Chen, Priya L Donti, Kyri Baker, J Zico Kolter, and Mario Bergés. 2021. Enforcing policy feasibility constraints through differentiable projection for energy optimization. In *Proceedings of the Twelfth ACM International Conference on Future Energy Systems*. 199–210.

[13] Dominik Danner, Robin Huwa, and Hermann De Meer. 2022. Multi-objective flexibility disaggregation to distributed energy management systems. *ACM SIGENERGY Energy Informatics Review* 2, 2 (2022), 1–12.

[14] Dominik Danner, Jan Seidemann, Michael Lechl, and Hermann de Meer. 2021. Flexibility disaggregation under forecast conditions. In *Proceedings of the Twelfth ACM International Conference on Future Energy Systems*. 27–38.

[15] Mohammad Dehghan, Mahmoud Zadehbagheri, Mohammad Javad Kiani, and Samad Nejatian. 2023. Virtual power plants planning in the distribution network constrained to system resiliency under extreme weather events. *Energy Reports* 9 (2023), 4243–4256.

[16] Sayed Yaser Derakhshandeh, Amir S Masoum, Sara Deilami, Mohammad AS Masoum, and ME Hamedani Golshan. 2013. Coordination of generation scheduling with PEVs charging in industrial microgrids. *IEEE Transactions on Power Systems* 28, 3 (2013), 3451–3461.

[17] Steven Diamond et al. 2016. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research* 17, 83 (2016), 1–5.

[18] Yang Gao et al. 2013. A contract-based approach for ancillary services in V2G networks: Optimality and learning. In *2013 Proceedings IEEE INFOCOM*. IEEE, 1151–1159.

[19] Logan Goldie-Scot. 2019. A behind the scenes take on lithium-ion battery prices. https://about.bnef.com/blog/behind-scenes-take-lithium-ion-battery-prices/

[20] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. 2018. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905* (2018).

[21] Bing Huang, Aart Gerard Meijssen, Jan Anne Annema, and Zofia Lukszo. 2021. Are electric vehicle drivers willing to participate in vehicle-to-grid contracts? A context-dependent stated choice experiment. *Energy Policy* 156 (2021), 112410.

[22] Shengyi Huang, Rousslan Fernand Julien Dossa, Chang Ye, Jeff Braga, Dipam Chakraborty, Kinal Mehta, and João G.M. Araújo. 2022. CleanRL: High-quality Single-file Implementations of Deep Reinforcement Learning Algorithms. *Journal of Machine Learning Research* 23, 274 (2022), 1–18. http://jmlr.org/papers/v23/21-1342.html

[23] Adugna Gebrie Jember et al. 2020. Game and contract theory-based energy transaction management for Internet of electric vehicle. *IEEE Access* 8 (2020), 203478–203487.

[24] Frank Kelly. 1997. Charging and rate control for elastic traffic. *European transactions on Telecommunications* 8, 1 (1997), 33–37.

[25] Tongxin Li, Bo Sun, Yue Chen, Zixin Ye, Steven H Low, and Adam Wierman. 2021. Learning-based predictive control via real-time aggregate flexibility. *IEEE Transactions on Smart Grid* 12, 6 (2021), 4897–4913.

[26] Fabio Lilliu, Torben Bach Pedersen, and Laurynas Šikšnys. 2021. Capturing Battery Flexibility in a General and Scalable Way Using the FlexOffer Model. In *2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE, 64–70.

[27] Yuan Liu, Mengmeng Tian, Yuxin Chen, Zehui Xiong, Cyril Leung, and Chunyan Miao. 2022. A contract theory based incentive mechanism for federated learning. In *Federated and Transfer Learning*. Springer, 117–137.

[28] Bijay Neupane, Laurynas Šikšnys, and Torben Bach Pedersen. 2017. Generation and evaluation of flex-offers from flexible electrical devices. In *Proceedings of the Eighth International Conference on Future Energy Systems*. 143–156.

[29] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).

[30] Andrew Patterson, Samuel Neumann, Martha White, and Adam White. 2023. Empirical Design in Reinforcement Learning. *arXiv preprint arXiv:2304.01315* (2023).

[31] Saidur Rahman et al. 2022. On Efficient Operation of a V2G-Enabled Virtual Power Plant: When Solar Power Meets Bidirectional Electric Vehicle Charging. In *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '22)*. ACM, 119–128.

[32] Saidur Rahman, Javier Sales-Ortiz, and Omid Ardakanian. 2023. Making a Virtual Power Plant out of Privately Owned Electric Vehicles: From Contract Design to Scheduling. In *Proceedings of the 14th ACM International Conference on Future Energy Systems*. 459–472.

[33] Nasrin Sadeghianpourhamami, Johannes Deleu, and Chris Develder. 2019. Definition and evaluation of model-free coordination of electrical vehicle charging with reinforcement learning. *IEEE Transactions on Smart Grid* 11, 1 (2019), 203–214.

[34] Nasrin Sadeghianpourhamami, Nazir Refa, Matthias Strobbe, and Chris Develder. 2018. Quantitive analysis of electric vehicle flexibility: A data-driven approach. *International Journal of Electrical Power & Energy Systems* 95 (2018), 451–462.

[35] Bernard Salanié. 2005. *The economics of contracts: a primer*. MIT press.

[36] Jonas Schlund, Marco Pruckner, and Reinhard German. 2020. Flexability-modeling and maximizing the bidirectional flexibility availability of unidirectional charging of large pools of electric vehicles. In *Proceedings of the eleventh ACM international conference on future energy systems*. 121–132.

[37] Laurynas Šikšnys, Torben Bach Pedersen, Muhammad Aftab, and Bijay Neupane. 2019. Flexibility modeling, management, and trading in bottom-up cellular energy systems. In *Proceedings of the Tenth ACM International Conference on Future Energy Systems*. 170–180.

[38] Emmanouil Valsomatzis, Torben Bach Pedersen, and Alberto Abelló. 2018. Day-Ahead Trading of Aggregated Energy Flexibility. In *Proceedings of the Ninth International Conference on Future Energy Systems* (Karlsruhe, Germany) *(e-Energy '18)*. Association for Computing Machinery, New York, NY, USA, 134–138. https://doi.org/10.1145/3208903.3208936

[39] Stijn Vandael, Bert Claessens, Damien Ernst, Tom Holvoet, and Geert Deconinck. 2015. Reinforcement learning of heuristic EV fleet charging in a day-ahead electricity market. *IEEE Transactions on Smart Grid* 6, 4 (2015), 1795–1805.

[40] Stijn Vandael, Bert Claessens, Maarten Hommelberg, Tom Holvoet, and Geert Deconinck. 2012. A scalable three-step approach for demand side management of plug-in hybrid vehicles. *IEEE Transactions on Smart Grid* 4, 2 (2012), 720–728.

[41] Matteo Vasirani et al. 2013. An agent-based approach to virtual power plants of wind power generators and electric vehicles. *IEEE Transactions on Smart Grid* 4, 3 (2013), 1314–1322.

[42] Jianing Wang, Chunlin Guo, Changshu Yu, and Yanchang Liang. 2022. Virtual power plant containing electric vehicles scheduling strategies based on deep reinforcement learning. *Electric power systems research* 205 (2022), 107714.

[43] Chenye Wu et al. 2011. Vehicle-to-aggregator interaction game. *IEEE Transactions on Smart Grid* 3, 1 (2011), 434–442.

[44] Zehui Xiong, Jiawen Kang, Dusit Niyato, Ping Wang, H Vincent Poor, and Shengli Xie. 2020. A multi-dimensional contract approach for data rewarding in mobile networks. *IEEE Transactions on Wireless Communications* 19, 9 (2020), 5779–5793.

[45] Haïkel Yaïche, Ravi R Mazumdar, and Catherine Rosenberg. 2000. A game theoretic framework for bandwidth allocation and pricing in broadband networks. *IEEE/ACM transactions on networking* 8, 5 (2000), 667–678.

[46] Ming Zeng et al. 2015. An incentivized auction-based group-selling approach for demand response management in V2G systems. *IEEE Transactions on Industrial Informatics* 11, 6 (2015), 1554–1563.

Algorithm 1: Priority Based Disaggregation

Inputs:
Total energy available y_t^{agg}
Individual upper bounds $\mathbf{y}_{t,upper} = \{y_{t,upper}^1, \ldots, y_{t,upper}^{N_t}\}$
Individual lower bounds $\mathbf{y}_{t,lower} = \{y_{t,lower}^1, \ldots, y_{t,lower}^{N_t}\}$
Priority of each EV $\mathbf{pr}_t = \{pr_t^1, \ldots, pr_t^{N_t}\}$

Output:
Energy allocated to each car $\mathbf{y}_t = \{y_t^1, \ldots, y_t^{N_t}\}$

1 $index_{pr} \leftarrow$ indices that would sort $\mathbf{pr}_t$ by descending;
2 $\mathbf{y}_t \leftarrow \mathbf{y}_{t,lower}$;
3 $y_{t,surplus}^{agg} \leftarrow y_t^{agg} - \sum(\mathbf{y}_{\mathbf{t,lower}})$;
4 **for** $i \leftarrow 1$ ***to*** $|\mathcal{N}_t|$ **do**
5 $\quad n \leftarrow index_{pr}[i]$;
6 $\quad y_{temp} \leftarrow \min(y_{t,surplus}^{agg}, (y_{t,upper}^n - y_{t,lower}^n))$;
7 $\quad y_t^n \leftarrow y_t^n + y_{temp}$;
8 $\quad y_{t,surplus} \leftarrow y_{t,surplus} - y_{temp}$;
9 $\quad$ **if** $y_{t,surplus}^{agg} == 0$ **then**
10 $\quad\quad$ **break**;
11 $\quad$ **end**
12 **end**
13 **return** $\mathbf{y}_t$;

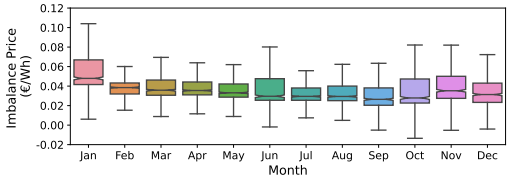

Figure 6: Distribution of prices in the imbalance market, whiskers show 1.5×IQR.

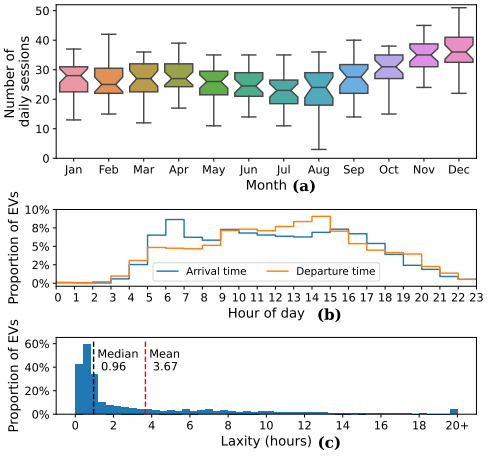

Figure 7: EV charging sessions dataset. Subplot (a) shows a boxplot of daily sessions per month, whiskers are 1.5×IQR. Subplot (b) shows a histogram of arrivals and departures. Subplot (c) shows a histogram of laxity at arrival.

A AGGREGATE STATE DEFINITION

The following features constitute the state of our RL agent. Firstly, we consider the features that describe EVs:

- Average upper bound; $\frac{1}{|\mathcal{N}_t|}\sum_{n\in\mathcal{N}_t} y_{t,upper}^n$.
- Average lower bound; $\frac{1}{|\mathcal{N}_t|}\sum_{n\in\mathcal{N}_t} y_{t,lower}^n/N_t$.
- Average minimum power to maintain non-negative laxity; $\frac{1}{|\mathcal{N}_t|}\sum_{n\in\mathcal{N}_t} y_{t,lower4}^n$
- Average state of charge of the EVs; $\frac{1}{|\mathcal{N}_t|}\sum_{n\in\mathcal{N}_t} soc_t^n$
- Average energy demand of the EVs.; $\frac{1}{|\mathcal{N}_t|}\sum_{n\in\mathcal{N}_t}\left(soc_{dep}^n - soc_t^n\right)$
- Average remaining time connected of the EVs.; $\frac{1}{|\mathcal{N}_t|}\sum_{n\in\mathcal{N}_t} \tau_t^n$
- Average laxity of the EVs.; $\frac{1}{|\mathcal{N}_t|}\sum_{n\in\mathcal{N}_t} lax_t^n$
- Proportion of connected cars with contracts; $|\mathcal{N}_t^D|/|\mathcal{N}_t|$;
- Average energy available for discharge in EVs with contracts; $\frac{1}{|\mathcal{N}_t^D|}\sum_{n\in\mathcal{N}_t^D} \bar{w}_t^n$
- Average remaining time of contracts; $\frac{1}{|\mathcal{N}_t^D|}\sum_{n\in\mathcal{N}_t^D} \bar{l}_t^n$

Note that for the features normalized by N_t, if $N_t = 0$ then the feature is set to 0. The same applies to the features normalized by N_t^D. In addition to these aggregate features, the agent also receives some additional features about the rest of the environment:

- One-hot encoded hour of day
- One-hot encoded day of the week
- Price predictions over the next h hours; $\hat{\mathbf{p}}_{t:t+h}$. Experimentally, we used $h = 8$.
- Discrete derivative of the price predictions; $\frac{\Delta\,\hat{\mathbf{p}}_{t:t+h}}{\Delta t}$
- Average slope of the price over the next h hours; $\frac{1}{h-1}\sum_{i=0}^{h-1}(\hat{p}_{t+1+i} - \hat{p}_{t+i})$

B ALGORITHM FOR PRIORITY BASED DISAGGREGATION

Algorithm 1 starts by calculating the indices that would sort the priority array, $\mathbf{pr}_t$, in descending order. These indices will allow us to loop through the other arrays from highest to lower priority. Line 2 sets the output vector $\mathbf{y}_t$ to $\mathbf{y}_{t,lower}$, because each EV must have at least its lower bound met. Next, in Line 3, we subtract the energy we just allocated from the total available. The result is stored in the new variable $y_{t,surplus}$.

We then enter the main loop. During the first iteration, we select, in Line 5, the index of the EV with highest priority, and the EVs with progressively lower priorities will be selected in later iterations. In Line 6, we check if we can assign all the surplus energy, $y_{t,surplus}$ to EV n. If not, we saturate it to its upper bound. In Line 7, that energy is allocated, and in Line 8 it is subtracted from the surplus. The loop breaks when all of the surplus energy has been allocated.

C DATASET VISUALIZATIONS

Fig. 6 shows the box plot of hourly prices throughout the year. The number of daily sessions is depicted in Fig. 7a. Histograms of arrival and departure times on each day are shown in Fig. 7b. Finally, Fig. 7c shows the histogram of laxity at arrival time.