EV2Gym: A Flexible V2G Simulator for EV Smart Charging Research and Benchmarking

Stavros Orfanoudakis *Student Member, IEEE*, Cesar Diaz-Londono, *Member, IEEE*, Yunus E. Yılmaz, Peter Palensky, *Senior Member, IEEE*, Pedro P. Vergara, *Senior Member, IEEE*

Abstract—As electric vehicle (EV) numbers rise, concerns about the capacity of current charging and power grid infrastructure grow, necessitating the development of smart charging solutions. While many smart charging simulators have been developed in recent years, only a few support the development of Reinforcement Learning (RL) algorithms in the form of a Gym environment, and those that do usually lack depth in modeling Vehicle-to-Grid (V2G) scenarios. To address the aforementioned issues, this paper introduces the EV2Gym, a realistic simulator platform for the development and assessment of small and large-scale smart charging algorithms within a standardized platform. The proposed simulator is populated with comprehensive EV, charging station, power transformer, and EV behavior models validated using real data. EV2Gym has a highly customizable interface empowering users to choose from pre-designed case studies or craft their own customized scenarios to suit their specific requirements. Moreover, it incorporates a diverse array of RL, mathematical programming, and heuristic algorithms to speed up the development and benchmarking of new solutions. By offering a unified and standardized platform, EV2Gym aims to provide researchers and practitioners with a robust environment for advancing and assessing smart charging algorithms.

Index Terms—Electric vehicle optimization, gym environment, reinforcement learning, mathematical programming, model predictive control (MPC).

I. INTRODUCTION

THE increasing number of electric vehicles (EVs) raises concerns about whether the current charging and power grid infrastructure can adequately accommodate them [1]. Therefore, it is essential to evaluate the readiness of the existing infrastructure and develop smart charging solutions using accurate simulator platforms. Various problem formulations related to EV smart charging exist, each with its own unique characteristics and constraints [2]. A variety of algorithms have been employed to tackle these challenges. Assessing the strengths and weaknesses of these solutions within a standardized simulator prior to real-world implementation can accelerate the integration of these methods into the practical operational management of EVs.

Classic mathematical optimization [3] algorithms and Model Predictive Control (MPC) [4] are effective for solving

less complex EV optimization problems. In particular, mixed integer programming (MIP) has gained widespread attention in the optimization of EV charging scheduling, as it can model the complexity of the problem, e.g., EVs arriving and departing, while taking into account various constraints [5], such as EV battery capacity, charging demand, and energy pricing [6]. However, mathematical programming encounters difficulties as the number of decision variables and constraints increases [7], for example, when Charge Point Operators (CPOs) need frequent rerunning of optimization algorithms. This operational demand, especially with the anticipated increase of EVs in the near future, poses a significant efficiency challenge for mathematical programming approaches when confronted with complex, large-scale optimization problems.

Reinforcement Learning (RL) methods [8] hold the potential to bridge the gap between optimality, linear modeling, uncertainty quantification, and scalability. In recent years, there has been a surge of interest in utilizing RL approaches to determine optimal EV charging behavior [9]. RL methods are capable of efficiently solving even the most complex problems, albeit sometimes at the expense of finding the optimal solution [10]. Therefore, the availability of standardized RL Gym environments for EV smart charging is crucial. Such environments can seamlessly support benchmarking for any type of open-source smart charging algorithm and facilitate the development of new specialized RL algorithms [11].

A. Existing EV Charging Simulators

The importance of developing and evaluating algorithmic solutions that can support the influx of hundreds of thousands of EVs in the coming years has led to the development of many EV simulator platforms. However, existing platforms are either outdated, w.r.t. the simulation models implemented (V2G, battery degradation, etc.) and the simulator, or have limited simulation capabilities. Table I provides a comprehensive comparison of existing EV simulator platforms, highlighting the advantages and the limitations of each one. The main comparison points revolve around the models developed, the EV behavior data, the study of the impact and the constraints of the power network, the developed algorithms, and their suitability for the development of RL algorithms.

V2G-Sim [12] is a traditional EV simulator with rich modeling capabilities, such as EV models, EV behavior, etc. Still, the simulator is not open-source and does not provide an environment for developing RL algorithms. EVLibSim [13] is another high-level optimal EV charging simulator for evaluating scenarios such as V2G, battery swapping, and inductive

TABLE I
OVERVIEW OF EXISTING EV SIMULATORS FOCUSING ON SMART CHARGING STRATEGIES.

Simulator Name	V2G	Power Network Impact	EV Models	EV Behavior	Charging Stations	Available Baseline Algorithms	RL Ready	Programming Language	Comments
V2G-Sim [12]	Yes	Partial	Diverse	Real Charging Transaction Probability Distributions	Uniform	- Heuristics, Mathematical Programming	No	Not Open-source	- Customizable V2G simulations.
EVLibSim [13]	Yes	No	Diverse	Randomized	Uniform	- Heuristics, Mathematical Programming	No	Java	- Easily customizable simulations with a visual interface.
EV-EcoSim [14]	Yes	Complete	Uniform	Randomized	Uniform	- Mathematical Programming	No	Python	- Grid-Impact analysis
evsim [15]	No	Partial	Diverse	Real Charging Transaction Probability Distributions	Uniform	- Heuristics, Mathematical Programming	No	R	- Simulate and analyze the charging behavior of EV users.
OPEN [16]	Yes	Complete	Uniform	Randomized	Uniform	- Heuristics, Mathematical Programming	No	Python	- Modelling, control & simulations for smart local energy systems.
ACN-Sim [17]	No	Complete	Uniform	Real Charging Transactions	Uniform	- Heuristics, MPC, RL, Mathematical Programming	Yes	Python	- Designing a complete simulator framework.
SustainGym [18]	No	Complete	Uniform	Real Charging Transactions	Uniform	- RL	Yes	Python	- Providing a benchmark for sustainable RL applications.
Chargym [19]	Yes	Very Limited	Uniform	Randomized	Uniform	- Heuristics, RL	Yes	Python	- Comparing RL algorithms for smart charging.
EV2Gym (Ours)	Yes	Partial	Diverse	Real Charging Transaction Probability Distributions	Diverse	- Heuristics, MPC, RL, Mathematical Programming	Yes	Python	- Comprehensive simulator for any control algorithm.

charging. This simulator focuses on providing diverse EV models but does not simulate the impact on the grid and by being written in Java it limits the accessibility to advanced machine learning packages developed in Python. Furthermore, EV-EcoSim [14] is an EV simulator environment focusing on the impact of charging on the distribution network. Additionally, it includes detailed battery utilization and degradation models but does not focus on realistic EV specification and behavior data. EVsim [15] is a simulator built with realistic EV behavior data from public charging transactions focusing on assessing and analyzing the behavior of many EV users. Meanwhile, it does not include the option to research V2G and its impact on the grid. Furthermore, OPEN [16] is an integrated modeling, control, and simulation framework for smart local energy systems, including EVs, also solving power flow at a distribution system level. However, it only has uniform EV specification and behavior models and is not suitable for the development of RL algorithms. Notably, these simulators lack the depth required for robust RL algorithm development due to either the absence of a standardized RL environment (Gym) or the oversimplified nature of their models.

To address this issue, most modern simulators also include a Gym environment that can seamlessly enable the testing of any RL algorithm. ACN-Sim [17] is one of the most established EV simulator platforms, as it is developed following real EV parking lot characteristics while also including a Gym environment. It is also one of the only EV-specific simulators supporting complete power network calculations using open-source grid simulators, such as Pandapower and MatPower. The main limitation of ACN-Sim is that it was not designed with V2G support in mind; hence, it is not suitable for V2G research. SustainGym [18] is an extension of ACN-Sim that tries to standardize the EV optimal charging problem as an RL benchmark suite by defining the state and action space; however, without adding substantial functionality to the simulation. Chargym [19] is another EV simulation environment that can be used to develop RL algorithms focused on cost and penalty design; however, the underlying models are very simplified. For example, all EVs and chargers have identical specifications, and the EV behavior is not based on real data. Overall, there are Gym environments that provide high-quality EV charging management simulation; however, each one has different areas of focus or is outdated. Therefore, it is important to develop a standardized platform that can support the development of any type of control algorithm and that is also capable of simulating V2G scenarios.

B. Our Contributions

To address these gaps, we introduce EV2Gym[1]. This innovative V2G simulator is tailored for developing and evaluating smart charging algorithms, including rule-based, mathematical programming, metaheuristic, and RL. Unlike existing tools, EV2Gym incorporates realistic CPO assumptions, a critical aspect often overlooked in other approaches. To enhance the fidelity of our simulator, we populate it with highly detailed models of EVs, charging stations, and EV behaviors using real open-source data. By offering a unified and standardized platform, EV2Gym aims to provide researchers and practitioners with a robust environment for the advancement and assessment of charging algorithms. The contributions of this paper can be summarized as:

- Flexible simulation environment for benchmarking and impact assessment of any charging algorithms.
- Detailed modeling of V2G EV charging management problems such as power setpoint tracking, profit maximization, utilization of PV, and demand response events.
- Fully customizable and easily configurable simulations allowing for highly detailed specifications of simulations with respect to charging topology and specifications, EV characteristics and behavior, prices, loads and generation on the level of the power transformer, etc.

II. THE EV2GYM SIMULATOR

EV2Gym is an open-source simulator environment for comprehensive V2G simulations focused on developing and
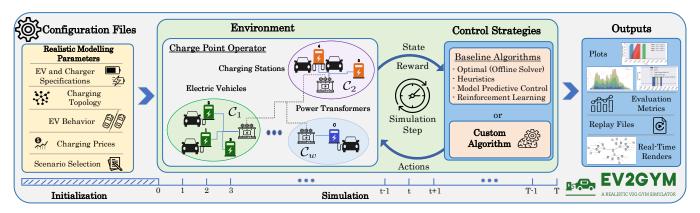
Fig. 1. An EV2Gym simulation comprises three phases: the configuration phase, which initializes the models; the simulation phase, which spans T steps, during which the state of models like EVs and charging stations is updated according to the decision-making algorithm; and finally, in the last phase, the simulator generates evaluation metrics for comparisons, produces replay files for reproducibility, and generates real-time renders for evaluation.
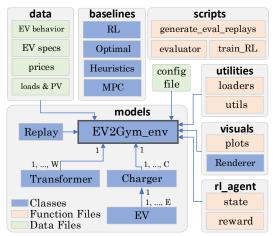

Fig. 2. Directory and file structure of the EV2Gym package.

Algorithm 1 Python code for running a simulation.

```
1: from EV2Gym.EV2Gym_env import EV2Gym
2: env← EV2Gym(config_file)              ▷ Initialization
3: state, _ ← env.reset()
4: agent ← Algorithm(...)        ▷ User-defined algorithm
5: for t = 1 to T do                     ▷ Simulation Phase
6:     actions ← agent.get_action(env, state)
7:     state, reward, done, _, stats ← env.step(actions)
8: end for
```

A. Simulation Flow & Configuration

Executing a simulation is straightforward, requiring just a few lines of code, as depicted in Alg.1. As a result, the simulation can be broken down into three phases, depicted in Fig. 1: Initialization, Simulation, and Evaluation. The first phase encompasses the initialization of simulator models, empowering users to modify simulation parameters (e.g., `EV2Gym(config_file)`), such as the simulation length T, timescale Δt, and charging topology configuration. Furthermore, the user can fully customize the parameters of individual simulation models, including EVs, chargers, and transformers. Additionally, users can integrate custom time-series data, e.g., electricity prices and inflexible load curves, enriching the simulation environment with real-world dynamics. Table II shows all the configuration parameters contained in a `config_file`.

Once the configuration phase has finished the main simulation takes place. The simulation is partitioned into discrete time steps represented by the set $\mathcal{T}$, where $t \in \mathcal{T}$. During a simulation, there is a fixed number of charging stations denoted by $\mathcal{C}$, where $i \in \mathcal{C}$, and is connected to a transformer $w \in \mathcal{W}$, while the number of the connected EVs changes dynamically based on the simulation time and the scenario. Additionally, each charging station has a set of Electric Vehicle Supply Equipment (EVSE) $\mathcal{J}$, where $j \in \mathcal{J}$, wherein each one an EV k from the set of available EVs $\mathcal{E}$ can connect. The goal of a simulation is to assess the performance of various charging strategies. Therefore, the user's charging strategy controls the current $I_{j,i,t}$ flowing from a charging station to a connected

evaluating charging strategies to assess their performance and limitations. Furthermore, EV2Gym integrates with the Gym API [11], streamlining the assessment of RL algorithms. The directory and file structure of the EV2Gym package is illustrated in Fig. 2, revealing the interconnection of its underlying classes, functions, and data files. The core component of EV2Gym is the `EV2Gym_env` class, which contains lists of associated entities such as `Transformers`, `Chargers`, and `EVs`. Additional essential functionalities are encapsulated within the `utilities`, `visuals`, and `rl_agent` directories. The `data` folder contains all the static data required for the simulation process. Furthermore, the `baselines` folder encompasses implementations of various smart charging algorithms, while the `scripts` directory holds utility functions for efficiently evaluating any case study.

Most importantly, EV2Gym is modular; thus allowing for the seamless addition of extra functionalities. This modularity is crucial, as open-source software empowers researchers to effortlessly extend the simulator to meet their specific requirements. This section introduces the components of EV2Gym and provides a detailed description of the simulation flow and its underlying models, i.e., EV, Charger, and Transformer.

Model	Input Parameters	Symbol
Simulation	- Timescale (in minutes)	Δt
	- Simulation Length (in steps)	T
	- Starting Date and Time	
	- Charging Topology and Properties	
	- EV Properties	
	- EV Scenario (Residential, Work, or Public)	
	- Set of EV Charging Stations	$\mathcal{C}$
	- Set of Power Transformers	$\mathcal{W}$
EV	- Min. & Max. AC Charging Power (kW)	$\underline{P}^{ch}_{AC}, \overline{P}^{ch}_{AC}$
	- Min. & Max. DC Charging Power (kW)	$\underline{P}^{ch}_{DC}, \overline{P}^{ch}_{DC}$
	- Min. & Max. Discharging Power (kW)	$\underline{P}^{dis}, \overline{P}^{dis}$
	- Min. & Max. Battery Capacity (kWh)	$\underline{E}, \overline{E}$
	- Charge & Discharge Efficiency	η^{ch}, η^{dis}
	- Battery Capacity at Arrival (kWh)	E^{arr}
	- Desired Capacity at Departure (kWh)	E^*
	- CV/CC Transition SoC (%)	τ
	- Time of Arrival & Time of Departure (t)	t^{arr}, t^{dep}
Charging Station	- Min. & Max. Charging Station Current (A)	$\underline{I}^{cs}, \overline{I}^{cs}$
	- Min. & Max. EVSE Charging Current (A)	$\underline{I}^{ch}, \overline{I}^{ch}$
	- Min. & Max. EVSE Discharging Current (A)	$\underline{I}^{dis}, \overline{I}^{dis}$
	- Voltage (V) & Phases	V, ϕ
	- Set of EVSEs	$\mathcal{J}$
	- Type of Charger (AC or DC)	
	- Charging & Discharging Prices (€/ kWh)	c^{ch}, c^{dis}
Transformer	- Min. & Max. Power (kW)	$\underline{P}^{tr}, \overline{P}^{tr}$
	- Inflexible Loads (kW)	P^L
	- Solar Power Generation (kW)	P^{PV}
	- Demand Response Event (kW)	P^{DR}
	- Set of Connected Charging Stations	$\mathcal{C}_w$

EV. For example, in a simulation with a set of charging stations $\mathcal{C}$ and each with a set of EVSEs $\mathcal{J}$ the control actions at timestep t is a vector $\boldsymbol{I}_t = [I_{j,i}], \forall j \in \mathcal{J}$ and $\forall i \in \mathcal{C}$.

As depicted in Alg. 1, a simulation step is split into taking an action based on a user-defined algorithm (agent), and then updating the state of the environment's models (env.step) conditioned on the action taken. In EV2Gym the control algorithm has complete access to the state of the environment, while it can also access future information, such as electricity prices and EV schedules (arrival, departure, and SoC). However, users can define which part of the input is observable based on the case study they are exploring. Most importantly, the EV2Gym is designed to support any type of algorithm, i.e., heuristics, RL, MPC, or mathematical programming.

Finally, the simulation ends after T steps have passed. At that time, the evaluation metrics (stats in line 6 of Alg. 1) of the simulation (see Sec. III-C), along with several figures, are generated in order to assess the performance of a charging algorithm. Moreover, a Replay is generated, storing information about the environment configuration parameters and the EV schedules, so that the same simulation can be evaluated with alternative smart-charging algorithms.

B. Electric Vehicles

Realistic EV models are necessary for precisely evaluating the effectiveness of charging strategies and the potential of

V2G technology in facilitating the energy transition. Consequently, the EV class is implemented to facilitate the simulation of diverse EV-related case studies.

1) Charging and Discharging Models: Similar to real cases, each EV can have different minimum and maximum charging and discharging power limits, depending on the charging mode (AC or DC), the power electronic limitations of the onboard battery management system (BMS), and the charger. In EV2Gym, the EV models have been designed with this property in mind. Thus, the min and max power limits are defined as $\underline{P}^{ch}$ and $\overline{P}^{ch}$, depending on the AC or DC charging mode. Similarly, the discharging power limits are defined as $\underline{P}^{dis}$, and $\overline{P}^{dis}$. Additionally, each EV has a maximum battery capacity, represented as $\overline{E}$, and a lower limit $\underline{E}$, which is used when discharging since the BMS of some EVs does not allow discharging under a certain threshold. Also, EVs have a desired SoC^* at departure.

In the EV2Gym, a configurable two-stage model is available. The charging and discharging power is

$$P_t = \eta \cdot I_t \cdot V_t \cdot \sqrt{\phi}, \tag{1}$$

where I_t is the control algorithm current, and η refers to the charging (η^{ch}) and discharging (η^{dis}) efficiency. Also, P_t depends on the charging station voltage V_t and the phases ϕ. P_t is also subject to the lower and upper power limits of the EV and the charging station. In detail, the two-stage model is:

$$SoC_t \begin{cases} SoC_{t-1} + P_t \cdot \Delta t / \overline{E} & SoC_{t-1} < \tau \\ 1 + (SoC_{t-1} - 1) \cdot exp\frac{P_t \cdot \Delta t}{E(\tau-1)} & SoC_{t-1} \geq \tau \end{cases} \tag{2}$$

where $\tau \in (0,1)$, is the SoC transition threshold signifying the start of the constant voltage region [17]. Notice that if $\tau = 1$, the model is linear. Fig. 3 demonstrates the charging and discharging curves of the two models compared to real lab measurements from [20]. As observed in Fig. 3, even though the current setpoint is practically constant the actual perceived current decreases after some point. The two-stage model can more accurately mimic the actual charging curve of Type-2 charging (Fig. 3a), while there is a slight deviation in the DC curves (Fig. 3b and Fig. 3c). These findings affirm the capability of EV2Gym models to accurately simulate realistic EV charging and discharging curves.

2) Battery Degradation: V2G holds promise in supporting power network operations through its ability to offer flexible loads and provide users with financial incentives. Nevertheless, there are concerns regarding the impact of discharging on EV batteries. To address this, the EV2Gym incorporates a validated battery degradation model [21], enabling the evaluation of various charging strategies. The battery degradation model has a calendar (d^{cal}) and a cyclic (d^{cyc}) capacity loss component. Capacity loss due to calendar aging over a simulation period T depends on the average $\langle SoC \rangle$ of the battery [22] and is defined as

$$d^{cal} = 0.75 \cdot (\epsilon_0 \cdot \langle SoC \rangle - \epsilon_1) \cdot \exp\left(-\frac{\epsilon_2}{\theta}\right) \cdot \frac{T}{(T^{tot})^{0.25}}, \tag{3}$$

where T^{tot} is the battery age in days, θ is the battery temperature, and $\epsilon_0, \epsilon_1, \epsilon_2$ are constants shown in Table III. The
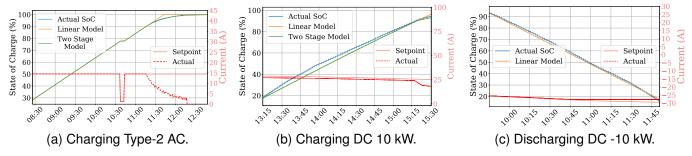
Fig. 3. Exploring the relation between actual and modeled SoC as a function of current in AC and DC charging and discharging.

TABLE III
BATTERY DEGRADATION PARAMETERS AND VALUES

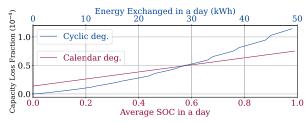| Param.	ϵ_0	ϵ_1	ϵ_2	θ	z_0	z_1	Q^{acc}
Value	$6.23 \cdot 10^6$	$1.38 \cdot 10^6$	6976	28	$4.02 \cdot 10^{-4}$	$2.04 \cdot 10^{-3}$	11160

Fig. 4. Visualizing calendar aging as a function of average SoC over a day, and cyclic degradation as a function of the total energy exchanged.

TABLE IV
TOTAL NUMBER OF REGISTERED EVS IN THE NETHERLANDS IN 2023

EV Model	Sales (2023 NL)	$\overline{E}$ (kWh)	$\overline{P}^{ch}_{AC}$ (kW)	$\overline{P}^{ch}_{DC}$ (kW)	$\overline{P}^{dis}_{DC}$ (kW)
Tesla Model 3	45545	57.5	11	170	-
Kia Niro	23105	64.8	11	80	-
Volkswagen ID.3	19950	58	11	120	-
Hyundai Kona	17752	64	11	77	-
Tesla Model Y	16186	57.5	11	170	-
Skoda Enyaq	16165	58	11	124	-
Peugeot 208	14017	46.3	7.4	101	-
Renault Zoe	14008	52	22	46	-
Volkswagen ID.4	13283	77	11	135	10
Volvo XC40	12520	66	11	135	-
Nissan Leaf	11977	39	3.6	46	7
Tesla Model S	10899	75	11	250	-

cyclic capacity loss depends on the total energy exchanged by the battery and the SoC at each simulation step

$$d^{cyc} = \left(\zeta_0 + \zeta_1 \frac{\int |\langle SoC \rangle - SoC(t)| \, \Delta t}{T} \right) \cdot \frac{\int |P(t)| \, \Delta t}{\sqrt{Q^{acc}}}. \quad (4)$$

In (4), the numerator of the last fraction represents the battery throughput during the simulation, while Q^{acc} represents the accumulated throughput during the battery's lifetime. The overall battery degradation is then defined as

$$Q^{lost} = d^{cal} + d^{cyc}. \quad (5)$$

The fraction of the capacity loss over a single day for an EV with a 2-year-old, 50 kWh battery is presented in Fig. 4. Battery degradation caused by calendar aging is highly dependent on the average SoC, while cyclic degradation can increase with the amount of energy exchanged.

3) EV Characteristics: EV2Gym includes realistic EV specifications. In detail, the user has the option to define the specifications of the EV models (see Table II) in a simulation either using realistic data from the RVO-NL [23] or custom ones. In cases where users opt for realistic EV specifications from the RVO-NL report, the simulator dynamically samples from the available EVs listed in Table IV, employing a probability distribution weighted by the total number of sales, each time a new EV is connected.

4) EV User Behavior Models: EV2Gym uses authentic EV behavior data to mimic various case studies. The simulation starts with no EVs connected, gradually introducing them in each time step t based on probability distributions for public, workplace, and residential load profiles provided by ElaadNL [24]. Users can specify the ElaadNL scenario to simulate, or can import custom EV behavior or charging transaction data from their private datasets. Specifically, whenever an EV is introduced at time t^{arr}, the EV2Gym utilizes these distributions to determine both the departure time t^{dep} and the energy level upon arrival E^{arr}, taking into account the time and day of arrival.

To validate the EV behavior model, we conducted extensive sampling, generating around 1 million charging transactions for each scenario. The distribution of arrival and departure times over a week is presented in Fig. 5, showcasing variations in daily routines between weekdays and weekends. Fig. 6 presents the probability density function (PDF) illustrating the duration of stay for each arrival time. Notably, in the workplace scenario, EVs do not arrive between 19:00-05:00, resulting in a zero stay time during these hours. Finally, Fig. 7 showcases the PDF of SoC upon arrival for every hour.

C. Charging Station

In EV2Gym, each `Charger` has a set of EVSE $\mathcal{J}$ where EVs can connect. Each charger can charge or discharge based on its voltage level V, number of phases ϕ^{cs}, and the total and EVSE current limitations (see Table II). In practice, the total current of a charging station is limited as

$$\underline{I}^{cs}_i \leq \sum_{j \in \mathcal{J}} I_{j,i} \leq \overline{I}^{cs}_i. \quad (6)$$

Additionally, if the current control signal of an EVSE is between; 0 and $\underline{I}^{ch}$ (for charging), or 0 and $\underline{I}^{dis}$ (for discharging),
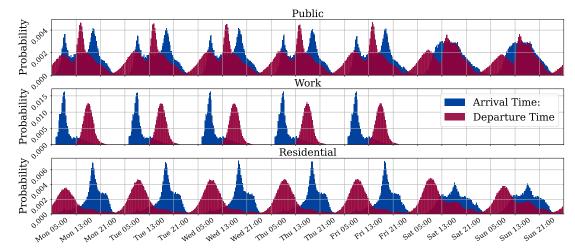
Fig. 5. Distributions of arrival and departure times of EVs as a function of time and scenario (public, work, and residential).
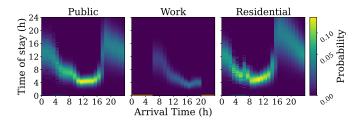

Fig. 6. Probability density function of the duration of stay for each hour of arrival in a day. The brighter the color the higher the probability that an EV spawns with that value. Notice that, all values for a single hour sum up to 1.
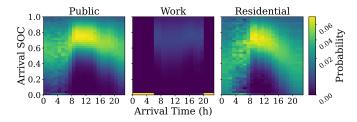

Fig. 7. PDF of the SoC at arrival for each hour of arrival in a day.

the actual current will be zero, because it is assumed that the EV's BMS will not allow for such low charging currents. Conversely, if the current signal exceeds the maximum EVSE current constraints, the BMS will limit the current to the maximum value. Also, if the sum of EVSE currents, $I_{j,i}$ for charger i, surpasses the charging station current limits, it is normalized down so that the operational constraint of (6) is not violated. Moreover, a charging station can have unique charging (c^{ch}) and discharging (c^{dis}) prices. EV2Gym uses by default the Dutch historic day-ahead energy prices, provided by entso-e [25], since they have a direct relation with the dynamic prices offered. If required, the user can include their own energy prices for their experiments.

D. Power Transformer

The current version of EV2Gym does not include power flow calculations; however, it aggregates sets of chargers $\mathcal{C}_w \subseteq \mathcal{C}$ at the level of power transformers $\forall w \in \mathcal{W}$.

As shown in Table II, a `Transformer` has power limits denoted by $\underline{P}^{tr}, \overline{P}^{tr}$. A power transformer in EV2Gym has inflexible loads (P^L) representing loads from houses or offices that cannot be shifted in time, solar power (PV) generation (P^{PV}), and dynamic capacity reduction events (P^{DR}), such as demand response events, that are communicated to the control algorithm only a few minutes ahead, e.g., 15 minutes ahead. The inflexible loads are based on randomized Pecan Street [26] datasets and the PV generation profile is based on randomized Renewables.ninja [27] data, while the occurrence of capacity reduction events is fully configurable by the user, with respect to start time, duration, etc. Therefore, the operational constraint for a power transformer w is defined as

$$\underline{P}^{tr}_w \leq P^{EVs}_w + P^L_w + P^{PV}_w \leq \overline{P}^{tr}_w - P^{DR}_w, \qquad (7)$$

where P^{EVs}_w is the total power because of EVs. Notice that, unlike the charging station that normalizes the excess current, violations of operational constraints of transformers are feasible. Overloads are measured throughout the simulations as they are an important aspect of a smart charging algorithm. EV2Gym also generates forecasts of inflexible loads ($\widetilde{P}^L$) and PV ($\widetilde{P}^{PV}$) by sampling from a Gaussian distribution $\mathcal{N}(\mu, \sigma)$ with mean (μ) the actual power consumed or generated at timestep t and standard deviation (σ) defined by the user.

III. EXPERIMENTAL DESIGN

This section showcases two common EV smart charging problems, modified through the simulation configuration file (`config_file`), to highlight the simulator's capabilities and inspire users to create their own case studies.

A. Power Setpoint Tracking

Scheduling and dispatching EVs often involves addressing challenges such as power setpoint tracking (PST), and capacity management. In practical scenarios, a CPO, or typically a company managing multiple EV chargers in their parking lot, either procures energy in advance from the day-ahead market or operates under a limited capacity contract. Consequently, they try to adhere closely to the procured power setpoint,

ensuring efficient charging for all connecting EVs while fairly distributing energy among them. In this scenario, we assume that information about EV arrival and departure time, and SoC at arrival is unavailable. However, it is assumed to be known when an EV is fully charged since the measured energy exchanged in a step is zero.

The PST problem has T discrete timesteps represented by the set $\mathcal{T}$, where $t \in \mathcal{T}$. There is a fixed number of charging stations denoted by $\mathcal{C}$, where $i \in \mathcal{C}$, is connected to a transformer $w \in \mathcal{W}$, while the number of EVs changes dynamically. To model the presence of an EV, a binary variable $u_{j,i,t}$ is introduced, with $u_{j,i,t} = 1$, indicating that an EV is connected and ready to charge at EVSE j at charging station i during time t. Therefore, the mathematical formulation leads to a MIP problem that is described by (8)-(22), $\forall w \in \mathcal{W}, \forall j \in \mathcal{J}, \forall i \in \mathcal{C}, \forall t \in \mathcal{T}$.

$$\min_{I_{j,i,t}^{ch}, I_{j,i,t}^{dis}} \sum_{t \in \mathcal{T}} \left(P_t^{set} - P_t^{tot}\right)^2 \tag{8}$$

Subject to:

$$P_t^{tot} = \sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{J}} \left(P_{j,i,t}^{ch} + P_{j,i,t}^{dis}\right) \qquad \forall j, \forall i, \forall t \tag{9}$$

$$P_{j,i,t}^{ch} = I_{j,i,t}^{ch} \cdot V_{j,i,t} \cdot \sqrt{\phi_{j,i,t}} \cdot \eta_{j,i,t}^{ch} \cdot \omega_{j,i,t}^{ch} \qquad \forall j, \forall i, \forall t \tag{10}$$

$$P_{j,i,t}^{dis} = I_{j,i,t}^{dis} \cdot V_{j,i,t} \cdot \sqrt{\phi_{j,i,t}} \cdot \eta_{j,i,t}^{dis} \cdot \omega_{j,i,t}^{dis} \qquad \forall j, \forall i, \forall t \tag{11}$$

$$\underline{E}_{j,i} \leq E_{j,i,t} \leq \overline{E}_{j,i} \qquad \forall j, \forall i, \forall t \tag{12}$$

$$E_{j,i,t} = E_{j,i,t-1} + \left(P_{j,i,t}^{ch} + P_{j,i,t}^{dis}\right) \cdot \Delta t \qquad \forall j, \forall i, \forall t \tag{13}$$

$$E_{j,i,t} = E_{j,i,t}^{arr} \qquad \forall j, \forall i, \forall t| \; t = t_{j,i,t}^{arr} \tag{14}$$

$$\underline{I}_{j,i}^{ch} \leq I_{j,i,t}^{ch} \leq \overline{I}_{j,i}^{ch} \qquad \forall j, \forall i, \forall t \tag{15}$$

$$\underline{I}_{j,i}^{dis} \geq I_{j,i,t}^{dis} \geq \overline{I}_{j,i}^{dis} \qquad \forall j, \forall i, \forall t \tag{16}$$

$$I_{i,t}^{cs} = \sum_{j \in \mathcal{J}} \left(I_{j,i,t}^{ch} \cdot \omega_{j,i,t}^{ch} + I_{j,i,t}^{dis} \cdot \omega_{j,i,t}^{dis}\right) \qquad \forall j, \forall i, \forall t \tag{17}$$

$$\underline{I}_i^{cs} \leq I_{i,t}^{cs} \leq \overline{I}_i^{cs} \qquad \forall i, \forall t \tag{18}$$

$$P_{w,t}^{EVs} = \sum_{i \in \mathcal{C}_w} \sum_{j \in \mathcal{J}} \left(P_{j,i,t}^{ch} + P_{j,i,t}^{dis}\right) \qquad \forall w, \forall j, \forall i, \forall t \tag{19}$$

$$\underline{P}_{w,t}^{tr} \leq P_{w,t}^{EVs} + P_{w,t}^L + P_{w,t}^{PV} \leq \overline{P}_{w,t}^{tr} - P_{w,t}^{DR} \qquad \forall w, \forall t \tag{20}$$

$$\omega_{j,i,t}^{ch} + \omega_{j,i,t}^{dis} \leq 1 \qquad \forall j, \forall i, \forall t \tag{21}$$

$$\omega_{j,i,t}^{ch} = \omega_{j,i,t}^{dis} = 0 \qquad \forall j, \forall i, \forall t| u_{j,i,t} = 0 \tag{22}$$

This formulation aims to minimize the squared power tracking error (8) by defining the charging and discharging current of the charging stations. Tracking error is the difference between the procured or setpoint power P_t^{set} and the actual power P_t^{tot} at time t. By minimizing this error, the costs of using unprocured energy and the losses of not using the procured energy are also minimized. The current of a single EVSE j is defined by two different decision variables $I^{ch} \cdot \omega^{ch}, I^{dis} \cdot \omega^{dis}$ to model the discharging behavior differently than the charging behavior. The charging current and power (I^{ch} and P^{ch}) are positive, while the discharging current and power (I^{dis} and P^{dis}) take negative values. (10) and (11) refer to the power definitions and (12)-(14) refer to the EV battery constraints. (15) and (16) represent current charging and discharging constraints for each EV and EVSE, and (18) for the whole charger. The transformer power constraint is described by (20). Finally, an EVSE cannot charge and discharge simultaneously; hence, the constraints in (21)-(22) for the binary variables ω^{ch} and ω^{dis}.

TABLE V
A LIST OF EVALUATION METRICS PROVIDED BY THE SIMULATOR.

Symbol	Metric	Equation
E^{ch}	-Total Energy Charged (kWh)	$\sum_{t \in \mathcal{T}} P_t^{ch} \cdot \Delta t$
E^{dis}	- Total Energy Discharged (kWh)	$\sum_{t \in \mathcal{T}} P_t^{dis} \cdot \Delta t$
ϵ^{usr}	- User Satisfaction (%)	$\frac{1}{|\mathcal{E}|} \cdot \sum_{k \in \mathcal{E}} \frac{SoC_k}{SoC_k^*}$
c	- Total Profits & Costs (€)	$\sum_{t \in \mathcal{T}} \left(P_t^{ch} \cdot c_t^{ch} + P_t^{dis} \cdot c_t^{dis}\right) \cdot \Delta t$
ϵ^{ov}	- Transformer Overload (kWh)	$\sum_{t \in \mathcal{T}} \sum_{w \in \mathcal{W}} max(P_{w,t}^{tr} - \overline{P}_{w,t}^{tr}, 0)$
$\epsilon^{|tr|}$	- Tracking Performance (kWh)	$\sum_{t \in \mathcal{T}} |P_t^{set} - P_t^{tot}| \cdot \Delta t$
ϵ^{tr}	- Squared Tracking Error	$\sum_{t \in \mathcal{T}} \left(P_t^{set} - P_t^{tot}\right)^2$
Q^{lost}	- Total Battery Capacity Loss	$\sum_{k \in \mathcal{E}} \left(d_k^{cal} + d_k^{cyc}\right)$

B. The V2G Profit Maximization Problem

The second problem investigated focuses on maximizing a CPO's profits while ensuring that EV users' demands are fully satisfied. In contrast to the PST problem, here, it is assumed that upon arrival at charging station i and EVSE j, an EV shares its time of departure ($t_{j,i}^{dep}$), and its desired battery capacity ($E_{j,i}^*$). Moreover, the battery capacity $E_{j,i,t}$ for each EV is known while it remains connected to a charger. These assumptions are usually considered in research as it is possible to retrieve this information from EVs, as more efficient communication protocols are introduced. The objective function is described in (23) as a function of the charging (c^{ch}) and discharging prices (c^{dis}) for each EVSE j, i.

$$\max_{I_{j,i,t}^{ch}, I_{j,i,t}^{dis}} \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{C}} \left(-P_{i,t}^{ch} \cdot c_{i,t}^{ch} + P_{i,t}^{dis} \cdot c_{i,t}^{dis}\right) \cdot \Delta t \tag{23}$$

Subject to constraints from (10)-(22) and:

$$E_{j,i,t} \geq E_{j,i,t}^* \qquad \forall j, \forall i, \forall t \,|t = t_{j,i,t}^{dep} \tag{24}$$

Moreover, in this scenario, the controller possesses knowledge of the inflexible load and PV forecasts to comply with the power transformer constraints (20).

C. Evaluation Metrics

At the end of a simulation, the list of evaluation metrics, shown in Table V is generated (see stats in Alg. 1). Evaluation metrics can help in assessing the performance of various smart charging algorithms uncovering their strengths and weaknesses. Also, more evaluation metrics can be added to fit the needs of the users.

IV. BASELINE ALGORITHMS

We offer a suite of baseline methods to expedite the comparison of smart-charging algorithms. These algorithms are categorized into heuristics, mathematical programming, and RL. This diverse selection allows for comprehensive evaluation across different algorithmic approaches.

A. Heuristics

Three simple rule-based algorithms are provided. The *Charge As Fast As Possible* (AFAP) algorithm charges EVs with maximum power immediately after they connect without complying with transformer-level constraints. Similarly, the *Charge As Late As Possible* (ALAP) heuristic starts charging EVs at maximum speed as late as possible to reach the desired SoC. Finally, *Round Robin* (RR) charges EVs in turns only up to the power setpoint (P^{set}), ensuring each EV gets a fair share of energy in a cyclical manner.

B. Mathematical Programming & MPC

Mathematical programming and Model Predictive Control (MPC) algorithms are also provided. In detail, when solving optimization problems, it is important to compare the outputs of the developed algorithms with optimal solutions. For this reason, we provide Gurobi models capable of solving optimally, assuming complete knowledge, the PST and V2G profit maximization problem, introduced in Sec. III. Moreover, real-time MPC solutions for the profit maximization problem are developed [28]. MPC is a dynamic approach, continuously recalibrating strategies to steer towards optimal outcomes, tackling the inherent uncertainties.

C. Deep Reinforcement Learning

Optimal EV dispatch problems are sequential decision-making problems under uncertainty; thus, they can be formulated as Markov Decision Processes (MDP) with $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$. The state-space $\mathcal{S}$ can include information about electricity prices, grid demand, user preferences, SoC, etc., and the actions $\mathcal{A}$ for every EV are related to the charging or discharging power during a time period t. The unknown state-transition probability is described by $\mathcal{P}$, while the reward $\mathcal{R}$ function can be designed to maximize any objective function related to EV optimization problems. Naturally, MDPs can be solved using RL [8] to learn the optimal policy by interacting with the environment and updating the policy based on observed rewards and states. EV2Gym is a Gym environment that can run any RL algorithm. In detail, we are utilizing the Stable Baselines 3 python package, which includes an implementation for the most common RL algorithms [29]: A2C, ARS, SAC, DDPG, TD3, TQC, PPO, TRPO, etc. Therefore, implementing custom algorithms, including designing new reward and state functions is simple to do. Furthermore, the simulator allows for the adjustment of the action space as needed, facilitating the implementation of discrete actions.

1) The PST Problem: Solving an EV optimal charging problem requires creating specific state and reward functions. In the case of PST problem, the state space is a vector consisting of $3+3|N|$ variables, where $|N|$ is the total number of EVSE where an EV can connect. In detail, at timestep t

$$S_t = [t, P_t^{set}, P_{t-1}^{tot}] \cup [d_{j,i}, E_{j,i,t} - E_{j,i}^{arr}, t - t_{j,i}^{arr}] \quad \forall j, \forall i \quad (25)$$

with $d_{j,i}$ being 1 if the EV is fully charged and 0.5 if the EV still receives energy. If no EV is connected at EVSE j of charging station i, then the EV state parameters are replaced

by zeros. Notice that in this case study we assume that we do not have prior information about EV arrivals and that the SoC is unknown because of the communication protocol. However, what is known is the total energy charged since the time of arrival ($E_{j,i,t} - E_{j,i}^{arr}$), and the time of stay up to step t. The reward function is $r_t = -\left(P_{t-1}^{set} - P_{t-1}^{tot}\right)^2$. To ensure bounded actions in RL, the action vector $\mathbf{a}$ is constrained within the interval $[0, 1]^N$, where a value of 1 indicates charging at maximum power and 0 denotes no action.

2) The V2G Profit Maximization Problem: Contrary to the realistic PST problem, here we assume we know information about EV departure and SoC as long as the EV connects to the charger. The charging energy prices, forecasted inflexible loads, and PV generation for a horizon h, are also considered to be known. Therefore, the state at timestep t is defined as:

$$S_t = [t, P_{t-1}^{tot}, c_{t:t+h}^{ch}] \cup [\widetilde{P}_{w,t:t+h}^L - \widetilde{P}_{w,t:t+h}^{PV}, P_{w,t:t+h}^{DR}]$$
$$\cup [SoC_{j,i}, t_{j,i}^{dep} - t] \quad \forall w, \forall j, \forall i. \quad (26)$$

The state has $1 + h + 2h|W| + 2|N|$ variables, where W is the total number of power transformers. This state space has many more variables than the PST problem, since the V2G problem is a more complex optimization task. The reward function $\forall w, \forall j, \forall i$ is:

$$r_t = c_{t-1} - 100 \cdot \epsilon_{w,t-1}^{ov} - 100 \cdot exp\left(-10 \cdot \epsilon_{j,i,t^{dep}}^{usr}\right), \quad (27)$$

where c_{t-1} are the costs, $\epsilon_{w,t-1}^{ov}$ are the power transformer overloads, and $\epsilon_{j,i,t^{dep}}^{usr}$ is the user satisfaction score of EVs at departure of the previous step, as defined in Table V. The function in (27) rewards profits while it penalizes heavily overloads and unsatisfied customers. The reward function in (27) was obtained after practical experimentation with alternative formulations. Similar to the PST problem, the action vector $\mathbf{a}$ is constrained within the interval $[-1, 1]^N$, where a value of -1 indicates discharging with maximum power, 1 means charging with maximum power and 0 denotes no action.

V. EXPERIMENTAL EVALUATION

This section presents an experimental evaluation of three case studies, each assessed using all available baseline algorithms and compared using the proposed evaluation metrics.

A. PST Benchmark

The first case study investigated is the public PST problem (see Sec. III-A) with 20 chargers, using 15-minute timesteps for almost a day (85 steps). Furthermore, the "Public" EV behavior model (Sec. II-B4) is employed, while the V2G functionality is not available, to replicate the charging behavior observed in publicly accessible charging stations. The primary difficulty of this case arises from the lack of prior information about EV arrivals and the absence of real-time SoC data. Table VI presents the average performance of various baselines after 100 stochastic experimental runs. Notice that, only the results from applicable baselines are presented because the ALAP, and MPC algorithms require prior information of the EV departure, which is assumed to be unknown to the CPO in this case study. A key metric is the energy error,

TABLE VI
BASELINE ALGORITHMS FOR THE PST PROBLEM WITH 20 CHARGERS.

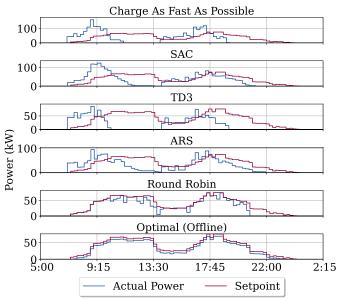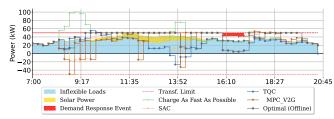| Algorithm	Energy Charged (kWh)	ϵ^{usr} (%)	Tracking Error (10^3)	Energy Error (kWh)	Reward (10^3)
AFAP	288 ±61	100 ±0	35.8 ±13.9	313 ±59	−18.2 ±7.5
RR	276 ±59	99 ±1	4.2 ±2.3	57 ±16	−2.0 ±0.8
A2C	144 ±41	84 ±4	30.5 ±12.0	295 ±57	−30.0 ±12.1
ARS	255 ±53	96 ±2	17.4 ±7.6	213 ±46	−15.8 ±7.3
SAC	281 ±59	99 ±1	20.1 ±8.6	231 ±48	−13.8 ±6.2
TD3	147 ±43	84 ±4	29.9 ±11.8	292 ±56	−29.4 ±11.8
TQC	252 ±54	96 ±2	20.8 ±8.8	237 ±49	−17.4 ±8.0
TRPO	79 ±30	77 ±4	25.2 ±11.6	264 ±60	−25.2 ±11.6
Optimal	287 ±60	100 ±0	0.6 ±0.3	46 ±14	−0.6 ±0.3

Fig. 8. Side-by-side comparison of baseline algorithms for the PST problem for 20 EV chargers throughout a day (85 steps, 15 minutes per step).

(a) Charging and discharging prices.

(b) Total power comparison.

Fig. 9. Performance comparison of selected algorithms in the V2G profit maximization with loads problem with 10 bidirectional single-EVSE chargers.

which indicates the actual deviation from the power setpoint. Additionally, metrics such as total energy charged, user satisfaction, tracking error, and reward are examined. The final row illustrates the optimal solution assuming complete knowledge, serving as an experimental upper bound on solution quality, although it does not represent a realistic scenario. Similarly, the AFAP heuristic algorithm serves as an experimental lower bound, illustrating the outcomes when no smart charging strategy is employed. Consequently, we anticipate that all other algorithmic solutions will yield results that are inferior to the optimal solution yet superior to those generated by AFAP.

As expected, AFAP has the worst average energy error (313 kWh) while RR has the best (57 kWh), excluding the optimal solution. The average energy error of RL baselines varies from 213 to 295 kWh, while the user satisfaction varies from 77% to 99%. RR also maximizes the fair distribution of energy as shown by the 99% user satisfaction. Even though RR seems to be the ultimate solution for the PST problem, it has limitations when the number of charging stations and operational constraints increase. Fig. 8 illustrates the performance of selected baseline algorithms for a single run.

B. V2G Profit Maximization with Loads Benchmark

The second case study is about maximizing the profits of V2G smart charging in a workplace parking lot while considering limited power capacity, uncertain inflexible loads, and PV generation. Here, we assume that EVs share information about their departure and the SoC is communicated at all times. The simulation uses the "Workplace" EV behavior model and has 85, 15-minute steps. Table VII demonstrates an extensive comparison of all evaluation metrics after 100 stochastic runs for every baseline algorithm. In this case study, profits and user satisfaction are the most important metrics. Here again, the optimal solution is not realistic but helps put an experimental upper bound. Notice that the total battery degradation of all EVs at the end of the simulations is measured for informational purposes, although it is not included in the objective or reward function of any model.

As shown in Table VII, no algorithm achieves 100% user satisfaction while maximizing profits. AFAP, ALAP, and RR do not consider the energy prices and V2G; hence, they fail to maximize profits. MPC, utilizing a prediction and control horizon of 25 steps, offers the most favorable balance between user satisfaction and profits. However, some RL methods, such as TD3, yield higher profits at the expense of reduced user satisfaction. Moreover, considering the execution time as a performance tradeoff is crucial. MPC requires significantly more time to determine optimal solutions, whereas RL algorithms are notably faster. The training time of RL is not considered in the comparison as it is done only once off-line. This underscores the limitations of mathematical programming solutions as problem complexity grows, thereby emphasizing the need for scalable solutions.

Fig. 9 illustrates the performance of selected algorithms in comparison to the charging and discharging prices. Utilizing uncontrolled charging (AFAP) results in higher transformer overloads, whereas MPC and optimal solutions efficiently utilize the power transformer limits to their maximum capacity. Conversely, RL approaches appear to be more conservative in approaching the transformer power limits.

Overall, in both case studies, the results suggest that RL

TABLE VII

BASELINES COMPARISON FOR THE V2G PROFIT MAXIMIZATION PROBLEM WITH 10 EV CHARGERS, TRANSFORMER LOADS, PV, AND DR EVENTS.

Algorithm	Profits/ Costs (€)	ϵ^{usr} (%)	Energy Ch. (kWh)	Energy Disch. (kWh)	Tr. Ov. (kWh)	Total Q^{lost} $(\times 10^{-3})$	$\sum d^{cal}$ $(\times 10^{-3})$	$\sum d^{cyc}$ $(\times 10^{-3})$	Execution Time (s)	Reward $(\times 10^{3})$
AFAP	−23.4 ±6.4	100 ±0	109 ±29	0 ±0	138 ±86	0.40 ±0.09	0.15 ±0.04	0.24 ±0.07	0.02 ±0.00	−13.8 ±8.6
ALAP	−25.2 ±7.1	100 ±0	109 ±29	0 ±0	106 ±78	0.38 ±0.09	0.14 ±0.04	0.24 ±0.07	0.02 ±0.00	−10.6 ±7.8
RR	−22.6 ±5.6	99 ±1	107 ±26	0 ±0	11 ±19	0.45 ±0.11	0.17 ±0.04	0.29 ±0.08	0.03 ±0.00	−1.2 ±1.9
ARS	17.7 ±9.9	53 ±11	45 ±19	107 ±34	36 ±67	0.57 ±0.14	0.12 ±0.03	0.45 ±0.11	0.09 ±0.01	−3.6 ±6.7
DDPG	25.3 ±9.5	44 ±9	33 ±16	125 ±34	43 ±88	0.59 ±0.13	0.12 ±0.03	0.47 ±0.11	0.12 ±0.02	−4.4 ±8.8
PPO	23.7 ±10.2	46 ±9	36 ±16	123 ±38	37 ±89	0.58 ±0.14	0.12 ±0.03	0.46 ±0.11	0.13 ±0.27	−3.7 ±8.9
TD3	31.1 ±10.6	38 ±8	25 ±13	142 ±41	74 ±140	0.61 ±0.13	0.12 ±0.03	0.50 ±0.11	0.12 ±0.03	−7.5 ±14.0
TQC	1.9 ±21.6	72 ±24	66 ±39	62 ±58	35 ±56	0.53 ±0.13	0.14 ±0.04	0.40 ±0.11	0.16 ±0.02	−3.5 ±5.6
TRPO	13.5 ±8.5	57 ±10	41 ±17	88 ±30	7 ±13	0.52 ±0.12	0.14 ±0.03	0.40 ±0.09	0.13 ±0.01	−0.7 ±1.3
MPC	17.8 ±9.3	76 ±9	480 ±107	462 ±109	183 ±167	1.68 ±0.42	0.12 ±0.03	1.55 ±0.39	108.18 ±19.44	−18.3 ±16.7
Optimal	3.7 ±6.1	100 ±0	578 ±135	469 ±117	4 ±12	1.45 ±0.35	0.15 ±0.04	1.31 ±0.32	33.30 ±28.05	−0.4 ±1.2

holds promise in offering effective solutions, albeit requiring some adjustments, such as modifying the reward or state functions, tuning algorithm hyperparameters, or exploring alternative algorithms. Nevertheless, EV2Gym stands ready to facilitate the development and comparative analysis of novel solutions, regardless of the approach chosen.

VI. CONCLUSIONS

In this paper, we introduced EV2Gym, an innovative V2G simulator designed to address critical gaps in the development and evaluation of smart EV charging algorithms. Unlike existing tools, EV2Gym integrates realistic CPO assumptions and detailed models of EVs, charging stations, and EV behaviors, leveraging real open-source data to enhance fidelity. Finally, we demonstrated the user-friendly process of designing new case studies or custom algorithms and leveraging existing baseline algorithms to accelerate the evaluation process.

REFERENCES

[1] H. Pandzic, B. Franc, S. Stipetic, F. Pandžić, M. Mesar, M. Miletić, and S. Jovanovic, "Electric vehicle charging infrastructure in croatia – first-hand experiences and recommendations for future development," *Journal of Energy - Energija*, vol. 71, pp. 16–23, 06 2023.

[2] H. S. Das, M. M. Rahman, S. Li, and C. W. Tan, "Electric vehicles standards, charging infrastructure, and impact on grid integration: A technological review," *Ren. and Sustainable Energy Reviews*, 3 2020.

[3] I. Sengor, O. Erdinc, B. Yener, A. Tascikaraoglu, and J. P. Catalao, "Optimal energy management of EV parking lots under peak load reduction based DR programs considering uncertainty," *IEEE Trans. on Sustainable Energy*, vol. 10, no. 3, pp. 1034–1043, 7 2019.

[4] C. Diaz-Londono, G. Fambri, P. Maffezzoni, and G. Gruosso, "Enhanced ev charging algorithm considering data-driven workplace chargers categorization with multiple vehicle types," *eTransportation*, vol. 20, 2024.

[5] A. M. Koufakis, E. S. Rigas, N. Bassiliades, and S. D. Ramchurn, "Offline and Online Electric Vehicle Charging Scheduling with V2V Energy Transfer," *IEEE Trans. on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 2128–2138, 5 2020.

[6] P. Meenakumar, M. Aunedi, and G. Strbac, "Optimal Business Case for Provision of Grid Services through EVs with V2G Capabilities," *Int. Conf. on Ecological Vehicles and Renewable Energies*, 9 2020.

[7] Y. Yang, H. G. Yeh, and R. Nguyen, "A Robust Model Predictive Control-Based Scheduling Approach for Electric Vehicle Charging With Photovoltaic Systems," *IEEE Systems Journal*, vol. 17, no. 1, 3 2023.

[8] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[9] D. Qiu, Y. Wang, W. Hua, and G. Strbac, "Reinforcement learning for electric vehicle applications in power systems:a critical review," *Renewable and Sustainable Energy Reviews*, vol. 173, p. 113052, 2023.

[10] X. Wang, S. Wang, X. Liang, D. Zhao, J. Huang, X. Xu, B. Dai, and Q. Miao, "Deep reinforcement learning: A survey," *IEEE Trans. on Neural Networks and Learning Systems*, pp. 1–15, 2022.

[11] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv:1606.01540*, 2016.

[12] S. Saxena, "Vehicle-to-grid simulator, version 00," 11 2013. [Online]. Available: https://www.osti.gov/biblio/1437011

[13] E. S. Rigas, S. Karapostolakis, N. Bassiliades, and S. D. Ramchurn, "EVLibSim: A tool for the simulation of electric vehicles' charging stations using the EVLib library," *Simulation Modelling Practice and Theory*, vol. 87, pp. 99–119, Sep. 2018.

[14] E. Balogun, E. Buechler, S. Bhela, S. Onori, and R. Rajagopal, "EV-EcoSim: A Grid-Aware Co-Simulation Platform for the Design and Optimization of Electric Vehicle Charging Infrastructure," *IEEE Trans. on Smart Grid*, pp. 1–1, 2023.

[15] M. Cañigueral, *evsim: Electric Vehicle Charging Sessions Simulation*, 2023, r package version 1.2.0. [Online]. Available: https://github.com/mcanigueral/evsim/

[16] T. Morstyn, K. A. Collett, A. Vijay, M. Deakin, S. Wheeler, S. M. Bhagavathy, F. Fele, and M. D. McCulloch, "Open: An open-source platform for developing smart local energy system applications," *Applied Energy*, vol. 275, p. 115397, 2020.

[17] Z. J. Lee, S. Sharma, D. Johansson, and S. H. Low, "ACN-Sim: An Open-Source Simulator for Data-Driven Electric Vehicle Charging Research," *IEEE Trans. on Smart Grid*, vol. 12, no. 6, Nov. 2021.

[18] C. Yeh, V. Li, R. Datta, J. Arroyo, N. Christianson, C. Zhang, Y. Chen, M. Hosseini, A. Golmohammadi, Y. Shi, Y. Yue, and A. Wierman, "SustainGym: Reinforcement learning environments for sustainable energy systems," in *NeurIPS Datasets and Benchmarks Track*, New Orleans, LA, USA, 12 2023.

[19] G. Karatzinis, C. Korkas, M. Terzopoulos, C. Tsaknakis, A. Stefanopoulou, I. Michailidis, and E. Kosmatopoulos, "Chargym: An EV Charging Station Model for Controller Benchmarking," in *Artificial Intelligence Applications and Innovations*. Springer International Publishing, 2022, pp. 241–252.

[20] E. Marcel, M. Marius, S. Alfio, and R. Christian, "Accurate ev charging profiles for simulation studies," Horizon 2020, EC grant agreement no 101056934, Lappeenranta/Lahti, Finland, D8.1, 2023.

[21] C. F. Lee, K. Bjurek, V. Hagman, Y. Li, and C. Zou, "Vehicle-to-grid optimization considering battery aging," *22nd IFAC World Congress*, vol. 56, no. 2, pp. 6624–6629, 2023.

[22] J. Schmalstieg, S. Käbitz, M. Ecker, and D. U. Sauer, "A holistic aging model for li(nimnco)o2 based 18650 lithium-ion batteries," *Journal of Power Sources*, vol. 257, pp. 325–334, 2014.

[23] Rijksdienst voor Ondernemend Nederland (RVO), "Statistics electric vehicles and charging in the netherlands: September 2023."

[24] "Elaadnl open datasets for electric mobility research," https://platform.elaad.io/analyses/ElaadNL_opendata.php.

[25] "Entso-e transparency platform," https://transparency.entsoe.eu/.

[26] "Pecan street data portal," https://www.pecanstreet.org/dataport/.

[27] S. Pfenninger and I. Staffell, "Long-term patterns of european pv output using 30 years of validated hourly reanalysis and satellite data," *Energy*, vol. 114, pp. 1251–1265, 2016.

[28] C. Diaz, F. Ruiz, and D. Patino, "Smart charge of an electric vehicles station: A model predictive control approach," in *2018 IEEE Conference on Control Technology and Applications (CCTA)*, 2018, pp. 54–59.

[29] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of ML Research*, vol. 22, no. 268, pp. 1–8, 2021.