38th International Electric Vehicle Symposium and Exhibition (EVS38)
Göteborg, Sweden, June 15-18, 2025

Adaptive Scheduling of Bidirectional EV Using SAC-Based Reinforcement Learning for Enhanced Grid Flexibility

Shiwei Shen[1], Niloofar Alirezai[1], Syed Irtaza Haider[1], Razan Habeeb[1]
Rico Radeke[1], Ralf Lehnert[1], Frank H. P. Fitzek[1,2]
[1]*Deutsche Telekom Chair of Communication Networks, TU Dresden, Germany*
shiwei.shen@tu-dresden.de
[2]*Centre for Tactile Internet with Human-in-the-Loop (CeTI)*

Abstract

The rise of electric vehicles (EVs) and photovoltaic (PV) systems advances renewable energy goals but also creates challenges due to the intermittent nature of PV generation. Bidirectional EV charging presents significant opportunities to enhance local energy resilience and to support the grid, where maintaining both positive and negative flexibility in real time is critical to accommodate unpredictable and dynamic grid imbalances.
This paper presents a smart charging solution based on Soft Actor-Critic (SAC), a deep reinforcement learning (DRL) method well-suited for managing continuous control tasks and promoting effective policy exploration. To ensure scalability across large EV fleets, we incorporate a Centralized Training with Decentralized Execution (CTDE) framework, which enables efficient learning and decentralized operation. The proposed approach optimizes charging costs, enhances PV energy utilization, and maintains flexibility to support grid operations. Simulation results demonstrate that our SAC-based framework consistently outperforms state-of-the-art DDPG-based methods across key operational metrics.

Keywords: Electric Vehicles, Modelling and Simulation, Smart charging, V2G

1 Introduction

The global shift toward environmental sustainability has accelerated the adoption of electric vehicles (EVs) and photovoltaic (PV) systems [1, 2], reflecting broader commitments to renewable energy and greenhouse gas reduction. However, the intermittent nature of PV generation often misaligns with local energy consumption, causing inefficiencies and underutilization. While unidirectional smart charging solutions address this by coordinating EV charging with PV production, recent attention has turned toward bidirectional charging for its greater potential to enhance energy system flexibility [3]. By enabling EVs to discharge energy back into local infrastructures, bidirectional charging improves the self-consumption of on-site renewable generation, strengthens local energy resilience, and reduces reliance on external grids. Furthermore, bidirectional EVs can contribute to broader grid stability through participation in ancillary services such as Frequency Containment Reserve (FCR) and Redispatch 3.0 [4, 5]. Despite its promise, the effective integration of bidirectional EVs into grid services remains challenging. EVs operating under smart charging plans may fail to absorb excess energy or deliver needed power during critical moments. As smart charging solutions often aim to maximize charging or discharging power to optimize utilization, this approach limits the ability of EVs to provide rapid, dynamic responses to unpredictable grid fluctuations, particularly amid the inherent uncertainty of grid demand and renewable

energy forecasts. To overcome these limitations, an effective bidirectional smart charging solution must not only optimize costs and PV utilization but also preserve both positive and negative flexibility at all times. By maintaining this dynamic readiness, EVs can enhance local renewable integration while actively supporting grid stability under real-time conditions.

While traditional optimization methods such as Mixed-Integer Linear Programming (MILP) and heuristic approaches have been widely applied to smart charging problems [3, 6], deep reinforcement learning (DRL) offers greater flexibility and scalability. By learning optimal policies through direct interaction with the environment, DRL is particularly well-suited to the high-dimensional, nonlinear, and stochastic dynamics characteristic of EV fleet management. Various DRL algorithms have been proposed for EV scheduling tasks [7–16]. These range from value-based methods, such as Deep Q-Networks (DQN) [11], which are more appropriate for problems with discrete action spaces, to policy gradient methods like Proximal Policy Optimization (PPO) [12] and Deep Deterministic Policy Gradient (DDPG) [13], which can handle continuous control tasks. In [12], the authors compare different DRL methods to solve a bidirectional EV scheduling problem and reduce charging costs. Their results show that DDPG can achieve higher rewards as opposed to PPO, but it also suffers from significant performance swings, while PPO shows better stability and, as a result, a more consistent performance. As shown in [14], a reinforcement learning framework for vehicle-to-building (V2B) strategies is proposed for smart communities with workplace EV charging, aiming to reduce peak energy costs and demand over extended periods. Their approach uses DDPG, enhanced with action masking and MILP-driven policy guidance, to handle complex factors like continuous action spaces and heterogeneous EV behavior. [15] explores a DDPG method to address the challenges of real-time charging control for EVs within smart grids. Their findings show that DDPG alone may not be the best-suited approach as it lacks sufficient exploration capabilities.

Soft Actor-Critic (SAC) [17] is an off-policy DRL method that combines continuous control with improved exploration through entropy maximization. The application of SAC to EV smart charging, especially for coordinated bidirectional management, remains relatively underexplored. It is particularly well-suited to bidirectional EV scheduling, where continuous action spaces are essential and adaptive responses to dynamic grid conditions are critical. However, as the number of EVs scales, the expansion of the state and action spaces significantly increases training time and complexity, posing an additional challenge. In this paper, we propose a smart charging framework based on SAC to enable flexible and adaptive EV scheduling in dynamic environments. To address the scalability challenges, we integrate a Centralized Training with Decentralized Execution (CTDE) framework, which supports efficient learning and decentralized operation. The key contributions of this work are as follows:

- We develop a smart charging framework based on SAC that satisfies drivers' needs while achieving system-wide objectives, including reducing charging costs, maximizing PV energy utilization, and enhancing grid flexibility.
- We adopt a CTDE paradigm, which reduces training time significantly by enabling decentralized decision-making while leveraging centralized information during training.
- We demonstrate the effectiveness of our approach through comprehensive simulations, comparing its performance against state-of-the-art DDPG-based methods with respect to charging costs, PV utilization, and operational constraints.

2 Method

In this section, the EV charging and discharging scheduling problem is modeled as a Markov Decision Process (MDP), which forms the foundation for the SAC algorithm. SAC is an off-policy, model-free DRL algorithm. SAC learns by interacting with the environment over time and optimizes both the expected cumulative reward and the entropy of the policy to encourage exploration.

2.1 SAC Method

The SAC framework is defined by the tuple $\langle \mathcal{S}, \mathcal{A}, p, r \rangle$, where:$\mathcal{S}$ is the continuous state space, $\mathcal{A}$ is the continuous action space, $p : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is the transition probability distribution, specifying the probability of moving to a new state given the current state and action, $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function, which provides feedback based on the current state and action. Unlike standard reinforcement learning objectives, SAC maximizes a maximum entropy objective, aiming to maximize both the expected reward and the entropy of the policy at each state. This leads to improved exploration and stability during training, with experiences stored in a replay buffer for efficient off-policy learning.

In SAC, the state space defines the environment, and the agent makes informed decisions after interacting with the environment. The overall state space is defined as:

$$s_t = [SoC_{n,t}, SoC^{dis}_{n,t}, \Delta SoC_{n,t}, E^{max}_n, T^{arr}_n, T^{dep}_n, \alpha_{n,t}, \beta_{n,t}, P_t, N_t, B_t, D_t, C_t, C^f_t, L^f_t, PV^f_t, D^f_t] \quad (1)$$

The state s_t defined in (1) consists of the following variables: $SoC_{n,t}$ denotes the State of Charge (SoC) of EV n at time t, calculated according to equation (2); $SoC_{n,t}^{dis}$ indicates the remaining discharge capacity of the battery, as computed in equation (3); $\Delta SoC_{n,t}$ represents the difference between the target state of charge and the current $SoC_{n,t}$ for EV n at time t; E_n^{max} is the maximum battery capacity of EV n; T_n^{arr} and T_n^{dep} denote the arrival and departure times of EV n, respectively; $\alpha_{n,t}$ is a binary flag indicating whether EV n is present at the charging station at time t; and $\beta_{n,t}$ represents the previous action taken for EV n. The global variables are defined as follows: P_t represents the cumulative power of all EVs at time t; N_t denotes the number of EVs present at the charging station; B_t indicates the cumulative action values from the previous timeslot; D_t is the power imbalance at time t, based on the load demand L_t and PV) generation PV_t. In addition, C_t represents the unit cost of grid energy at time t. C_t^f, L_t^f, PV_t^f, and D_t^f correspond to the unit cost of energy, load demand, available PV power, and power imbalance over the next 24 timeslots.

$$SoC_{n,t} = SoC_{n,t-1} + \frac{\tau \cdot \max(P_{n,t-1}, 0)\, \eta^C}{E_n^{\max}} + \frac{\tau \cdot \min(P_{n,t-1}, 0)}{E_n^{\max}\, \eta^D} \tag{2}$$

$$SoC_{n,t}^{dis} = SoC_n^{\max,dis} + \frac{\tau \cdot \sum_{i=T_n^{arr}}^{t} \min(P_{n,i}, 0)}{E_n^{\max}\, \eta^D} \tag{3}$$

In our simulation, the maximum discharge capacity of EV n is denoted as $SoC_n^{\max,dis}$, which is 50% of the battery capacity. The charging and discharging efficiencies are represented by η^C and η^D, respectively. The parameter τ denotes the duration of each timeslot.
The continuous action space defines the possible actions the agent can take, where the action for each EV n at time t is within the range $a_{n,t} \in [-P_{\max}, P_{\max}]$. The agent has access to the entire state space and independently decides on charging or discharging actions for each EV, with negative values indicating discharging and positive values indicating charging. Actions in the range $[-4\,\text{kW}, 4\,\text{kW}]$ are masked to 0, as the charging and discharging efficiency is low within this range.

$$\mathbf{a_t} = [a_{1,t}, a_{2,t}, \ldots, a_{N,t}] \tag{4}$$

In our problem formulation, after the agent selects the actions for all EVs based on the current policy, the environment evaluates these actions and provides a reward. The reward reflects how well the agent's decisions align with the system's overall objectives and constraints. This feedback guides the agent in improving its policy by maximizing the reward over time.

The total reward for timestep t is given by:

$$r_t = r_t^{\text{EV}} + r_t^{\text{global}} + r_t^{\text{obj}} \tag{5}$$

where: r_t^{EV} represents the EV-related reward, r_t^{global} represents the global reward, r_t^{obj} represents the objective reward. The first term, r_t^{EV}, accounts for violations related to individual EVs. This includes penalties for SoC violations, final SoC target violations, and discharge limit violations. The total EV-related reward is given by:

$$r_t^{\text{EV}} = -\frac{1}{N} \sum_{n=1}^{N} \left(\omega_{\text{soc}} \cdot r_{n,t}^{\text{SoC}} + \omega_{\text{dis}} \cdot r_{n,t}^{\text{dis}} + \omega_{\text{final}} \cdot r_{n,t}^{\text{final}} \right) \tag{6}$$

where $\omega_{\text{soc}}, \omega_{\text{dis}}, \omega_{\text{final}}$ are the weights of the penalties associated with SoC violation, discharge limit violation, and final SoC target violation, respectively. To promote feasible and user-compliant charging behaviors, we impose penalties for deviations below the minimum SoC and for overcharging beyond 100% SoC, as defined by the following penalty for SoC violations:

$$r_{n,t}^{\text{SoC}} = \begin{cases} 1 & \text{if SoC}_{n,t} < 0.2 \text{ or SoC}_{n,t} > 1 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

To protect battery health, we impose a penalty for discharge limit violations, as given by the following equation:

$$r_{n,t}^{\text{dis}} = \begin{cases} 1 & \text{if SoC}_{n,t}^{\text{dis}} < 0 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

To meet user requirements, it is important for the EVs to reach their target SoC before departure. The penalty for final SoC target violations is given by:

$$r_{n,t}^{\text{final}} = \begin{cases} 1 & \text{if } t = T_n^{\text{dep}} - 1 \text{ and } \text{SoC}_{n,t} < \text{SoC}_n^{\text{target}} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

The second term, r_t^{global} in equation 5, accounts for global violations across all EVs in the charging system. This includes penalties for fuse violations, peak violations, and over-discharge penalties. The total global reward is given by:

$$r_t^{\text{global}} = -\omega_{\text{fuse}} \cdot r_t^{\text{fuse}} - \omega_{\text{peak}} \cdot r_t^{\text{peak}} - \omega_{\text{overdis}} \cdot r_t^{\text{overdis}} \tag{10}$$

where $\omega_{\text{fuse}}, \omega_{\text{peak}}, \omega_{\text{overdis}}$ are the weights of the penalties associated with fuse violation, peak violation, and over-discharge violation, respectively. In order to ensure the safety and reliability of the charging station infrastructure, the penalty for fuse violations is defined as:

$$r_t^{\text{fuse}} = \begin{cases} 1 & \text{if } \sum_{n=1}^{N} P_{n,t} > N \cdot P^{\text{fuse}} \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

To meet the peak shaving target, a penalty is applied for peak violations as follows:

$$r_t^{\text{peak}} = \begin{cases} 1 & \text{if } \sum_{n=1}^{N} P_{n,t} + D_t > N \cdot P^{\text{peak}} \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

In order to prevent energy waste and ensure that only the necessary amount of energy is discharged, the penalty for over-discharge is defined as:

$$r_t^{\text{overdis}} = \begin{cases} 1 & \text{if } \sum_{n=1}^{N} P_{n,t} < 0 \text{ and } \left(D_t + \sum_{n=1}^{N} P_{n,t}\right) < 0 \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

The third term, r_t^{obj} in equation 5, represents the optimization objectives, which include minimizing the charging costs, maximizing PV utilization, and promoting system flexibility. The total objective reward is given by:

$$r_t^{\text{obj}} = -\omega_{\text{cost}} \cdot r_t^{\text{cost}} - \omega_{\text{PV}} \cdot r_t^{\text{PV}} - \omega_{\text{flex}} \cdot r_t^{\text{flex}} \tag{14}$$

The penalty for charging cost aims to reduce the overall cost of buying energy from the grid:

$$r_t^{\text{cost}} = C_t \cdot \tau \cdot \max\left(\sum_{n=1}^{N} P_{n,t} + D_t, 0\right) \tag{15}$$

The penalty for PV utilization encourages maximizing the use of available PV energy:

$$r_t^{\text{PV}} = \sum_{n=1}^{N} \min\left(P_{n,t}, \max(PV_t - L_t, 0)\right) \tag{16}$$

The penalty for system flexibility is:

$$r_t^{\text{flex}} = \frac{1}{N} \sum_{n=1}^{N} \left(\max\left(|P_{n,t}| - m_{\text{flex}}, 0\right)\right)^2 \tag{17}$$

The weight factors used in the reward equations are listed in Table 1.

2.2 CTDE-SAC

With the increase in the number of EVs in the schedule, the state and action dimensions grow accordingly (as shown in equations (1) and (4). As a result, SAC encounters difficulties in effectively handling this enlarged state-action space. In particular, SAC struggles with distortions that arise in high-dimensional continuous action spaces [18], which can lead to inaccuracies in action selection, value estimation, and high computation time. With the growing environment, these issues can affect stable policy learning. To overcome the scalability limitations of standard SAC, we use the CTDE approach and modify the architecture by employing a single centralized agent that selects actions for each EV through a sequential for loop. To implement the CTDE-SAC method, the state space in (1) needs to be separated into two parts, local and global. The local state contains only EV-specific information, while the global state space excludes any EV-specific variables; it instead represents shared common environmental information used for decision making across all EVs. The local state of EV_n and global state at timeslot t are defined in equations (18) and (19), respectively.

$$s_{n,t}^{l} = [SoC_{n,t}, SoC_{n,t}^{dis}, \Delta SoC_{n,t}, E_n^{max}, T_n^{arr}, T_n^{dep}, \alpha_{n,t}, \beta_{n,t}] \tag{18}$$

$$s_t^{g} = [P_t, N_t, B_t, D_t, C_t, C_t^f, L_t^f, PV_t^f, D_t^f] \tag{19}$$

To enable effective training, the SAC framework with a centralized training and decentralized execution is applied. As shown in the algorithm for CTDE-SAC in Table. 1, the system utilizes a single shared actor network for all agents, along with two separate critic networks (Q_{θ_1} and Q_{θ_2}) for estimating the Q-values. As depicted in Table. 1, the initialization step involves setting up the actor, critic, and target networks, as well as the replay buffer. In each episode step, when choosing actions, the actor receives both the global s_t^g and the local state $s_{n,t}^l$ for each EV_n at time t. After all the actions are computed and put in a joint action vector $\mathbf{a_t}$ shown in (4), the action chosen for absent EVs at time will be masked to 0. Only after all the actions are computed, the environment provides feedback in the form of reward denoted as r_t and calculated by (5) and transitions to the next state s_{t+1}. The relevant experiences are then stored in a replay buffer $\mathcal{D}$. States denoted by s_t and s_{t+1} are calculated based on (1) and contain the local information for all EVs as well as global information.

During training, a batch of transitions $(s_t, \mathbf{a_t}, r, s_{t+1})$ are sampled from the replay buffer. With this batch, the first step is to update the critic networks. To compute the target Q-value, the next action vector $\mathbf{a}_{t+1}$ is constructed by looping through each EV. In each iteration, the actor receives the corresponding local state $s_{n,t+1}^l$ along with the global state s_{t+1}^g and produces an action $a_{n,t+1}$ and its log probability $\log \pi(a_{n,t+1} \mid s_{n,t+1}^l)$. These are concatenated to form the full action vector, and the total log probability is accumulated. The target Q-value is then computed by taking the minimum between the two target Q-networks and subtracting the entropy term as shown in (20).

$$y_t = r_t + \gamma(1 - \text{done}) \left(\min_{j=1,2} Q_{\theta_j'}(s_{t+1}, a_{t+1}) - \alpha \sum_n \log \pi(a_{n,t+1} \mid s_{n,t+1}) \right) \tag{20}$$

Here, γ is the discount factor, α is the temperature parameter, and π is the policy probability for action a_n given the observation of $s_{n,t+1}$, and done if $True$ indicates the end of an episode. Next, the current Q-values are computed using the main critic networks Q_{θ_1} and Q_{θ_2} on the sampled state-action pairs $(s_t, \mathbf{a_t})$. The critic loss is then calculated using the mean squared error between current Q-values and the target Q-value. The critics are then updated using the loss function in (21).

$$\mathcal{L}_{Q_{\text{critic}}} = \mathbb{E} s_t, \mathbf{a_t} \left[\left(Q_{\theta_1}(s_t, \mathbf{a_t}) - y_t\right)^2 + \left(Q_{\theta_2}(s_t, \mathbf{a_t}) - y_t\right)^2 \right] \tag{21}$$

Once the critic networks are updated, the actor is also updated using the sampled batch. For each iteration n, the actor takes in its observation $s_{n,t}$ and outputs an action $a_{n,t}$ and the corresponding log-probability $\log \pi_\phi(a_{n,t} \mid s_{n,t})$. These actions are combined into the joint action vector $\mathbf{a_t}$. The update rule for the actor is derived from equation (22).

$$\nabla_\phi J = \mathbb{E} \left[\sum_n \left(\alpha \nabla_\phi \log \pi_\phi(a_{n,t} \mid s_{n,t}) - \nabla_\phi Q(s_{n,t}, a_{n,t}) \right) \right] \tag{22}$$

The gradient helps the actor balance between maximizing the Q-value and maintaining enough exploration, ensuring both improved performance and adaptability. The actor is optimized to increase the expected return while ensuring sufficient stochasticity in the policy.

Additionally, the temperature parameter α is automatically tuned to match a target entropy. The target entropy encourages exploration by penalizing low-entropy (overly deterministic) policies using (23).

$$\mathcal{L}(\alpha) = \mathbb{E}_{\mathbf{a_t} \sim \pi_t} \left[-\alpha \log \pi_t(\mathbf{a_t}|s_t) - \alpha \mathcal{H}_{\text{target}} \right] \tag{23}$$

Finally, the target Q-networks are updated using a soft update mechanism in (24).

$$\theta'_j \leftarrow \tau \theta_j + (1 - \tau_s)\theta'_j \tag{24}$$

where τ_s is a small constant that controls the update rate, ensuring stable target estimates.

Table 1: CTDE-SAC for EV charging

Algorithm: Centralized Training with Decentralized Execution (CTDE-SAC) for EV Charging

Input: PV, load, electricity cost, and hyperparameters such as: α, γ, learning rates, batch size, replay buffer size, etc.

Initialize:
  Shared actor network π_ϕ for all EVs
  Critic networks $Q_{\theta_1}, Q_{\theta_2}$
  Target networks $Q_{\theta'_1}, Q_{\theta'_2}$
  Replay buffer $\mathcal{D} \leftarrow \emptyset$
for each training episode **do**
  Reset environment
  for each timestep t until episode ends **do**
    # Decentralized Execution
    for each EV n **do**
      Get local and global state $s_{n,t}$ (e.g., SoC, arrival time)
      Select action $a_{n,t} = \pi_\phi(s_{n,t}) + \epsilon$
    end for
    Mask inactive EVs: $a_{n,t} \leftarrow 0$ if EV n not present
    Execute joint action $\mathbf{a}_t = [a_{1,t}, \ldots, a_{N,t}]$
    Observe next state s_{t+1}, reward r_t, and done signal
    Store relevant experiences $(s_t, \mathbf{a}_t, r_t, s_{t+1}, \text{done})$ in $\mathcal{D}$
    # Centralized Training
    if training step **then**
      Sample batch $\{(s_t, \mathbf{a_t}, r_t, s_{t+1}, \text{done})\} \sim \mathcal{D}$
      for each EV i in batch **do**
        Get local and global state $s_{n,t+1} \leftarrow \text{GetLocalandGlobalObs}(s_{t+1})$
        Compute $a_{n,t+1}, \log \pi \leftarrow \pi_\phi(s_{n,t+1})$
      end for
      Execute joint next action $\mathbf{a}_{t+1} = [a1, t+1, \ldots, a_{N,t+1}]$
      Calculate target Q-value using Eq. (20)
      Update critics using Eq. (21)
      Update shared actor policy π_ϕ using policy gradient Eq. (22)
      Update temperature parameter α to match target entropy using Eq. (23)
      Update target networks using Eq. (24)
    end if
  end for
end for
Output: Shared actor policy π_ϕ for decentralized execution

3 Results

This study investigates a corporate parking lot with 25 company-owned EVs and charging stations, operating within a power range of 4 to 22 kW in both directions. We also incorporate PV panels, with generation data based on summer 2023 weather in Dresden, Germany, updated every 15 minutes to reflect changing conditions. Electricity prices in 2023 are based on historical data from Germany's energy market. We simplify the pricing model by assuming a constant price for each 15-minute timeslot, serving as input for the DRL model. The load data was obtained from a municipal hospital in Dresden for 2023. The key simulation parameters, including the EV-related values and the CTDE-SAC hyperparameters, are outlined in Table 2.

Table 2: Simulation parameters for EV charging and CTDE-SAC model

Parameter	**Symbol**	**Values**
Number of EVs	N	[5, 25]
Number of Episodes	ε	4000
Time Slots	T	96
Battery Capacity	B_{capacity}	85 kWh
Charging/Discharging Power	$P^{\text{ch/dis}}$	22 kW
Efficiency (Charging/Discharging)	η^C/η^D	0.95
EV Arrival Time	T^{arr}	$\mathcal{N}(8, 0.5^2)$ (bounded [7 - 9] hrs)
EV Departure Time	T^{dep}	$\mathcal{N}(16, 0.5^2)$ (bounded [15 - 17] hrs)
Arrival SoC	SoC^{arr}	$\mathcal{N}(0.4, 0.1^2)$ (bounded [0.3 - 0.6])
SoC Weight	ω_{SoC}	50
Discharge Weight	ω_{dis}	20
Final SoC Weight	ω_{final}	100
Fuse, Peak, Overdischarge Weight	$\omega_{\text{fuse}}, \omega_{\text{peak}}, \omega_{\text{overdis}}$	100
Cost, PV, Flex Weight	$\omega_{\text{cost}}, \omega_{\text{PV}}, \omega_{\text{flex}}$	0.01
Discount Factor	γ	0.99
Learning Rates (Actor/Critic)	-	8×10^{-3}
Net Width (Actor/Critic)	W_{net}	2 layers of 256 units
Batch Size	B_{batch}	512
Replay Memory Size	M_{replay}	1×10^6
Training Frequency	f_{train}	1
Evaluation Frequency	f_{eval}	5

As previously discussed, standard SAC can struggle with high-dimensional state and action spaces. To validate this, we first evaluated our environment using a standard single-agent SAC model. After 8 hours of training, the model failed to meet the termination criteria and continued to exhibit a high number of constraint violations. These results demonstrate the limitations of standard SAC in complex environments, and clearly support the choice of CTDE-based methods for EV scheduling.

Table 3 presents a comparison of three methods: CTDE-DDPG, CTDE-SAC, and CTDE-SAC with Flexibility, evaluated under two configurations: 5 EVs and 25 EVs. The key metrics evaluated include cost per kWh, violation count, computation time, and the ability to meet the target SoC. A termination criterion was embedded in our environment to end the training process if an episode had no EV-related or fuse violations and up to 2 peak violations.

Regarding cost per kWh, the minimum, maximum, and mean values during the period from the arrival of the first EV to the departure of the last EV are 0.0030 €/kWh, 0.1082 €/kWh, and 0.0569 €/kWh, respectively. For the 5 EV scenario, CTDE-DDPG incurs the highest cost at 0.0538 €/kWh, followed closely by the proposed method at 0.0532 €/kWh. In contrast, for the 25 EV scenario, the proposed method has the highest cost at 0.0481 €/kWh, slightly higher than CTDE-SAC (0.0469 €/kWh), while CTDE-DDPG results in the lowest cost (0.0428 €/kWh). However, CTDE-DDPG's lower cost can be attributed to its inability to meet the required SoC, which results in a lower overall charging demand and consequently reduced cost. As the number of EVs increases from 5 to 25, the flexibility in managing the EV charging load through smart charging strategies helps reduce grid demand, effectively lowering the overall cost by optimizing the charging of the large EV fleet. It is important to note that the building

load remains the same in both the 5 EV and 25 EV scenarios, so the cost reduction is largely driven by managing the charging of flexible EV load.

For the 5 EVs scenario, the proposed method with flexibility achieved 94% PV utilization, absorbing more excess PV compared to CTDE-DDPG, which achieved 83%. The CTDE-SAC method without flexibility achieved 98%. In the 25 EVs scenario, all three methods reached 100% PV utilization, as the excess PV remained the same with the increase in the number of EVs.

The violation count, including SoC violations (where SoC should not fall below 0.2 or exceed 1.0), discharge limit violations (where the EVs should not discharge more than 50% of their battery capacity), fuse violations (related to the total charging demand exceeding the fuse limit), and peak violations (where demand from the grid should not exceed a peak shaving target), varies significantly across methods. CTDE-DDPG suffers from the highest violation count in both scenarios. In contrast, the proposed method demonstrates better performance, with only 3 peak violations in the 5 EV scenario and just 1 SoC violation in the 25 EV scenario, while CTDE-DDPG suffers from 23 SoC violations. This indicates that CTDE-DDPG struggles to manage SoC levels across a larger number of EVs, whereas the proposed method more effectively balances charging demands, reducing violations. In practical implementations, these violations can be eliminated using rule-based methods.

The computation time in Table 3 is measured from the start of training until the termination criterion is met. The proposed method consistently outperforms CTDE-DDPG in terms of computation time, achieving 0.53 hours for the 5 EV scenario and 0.78 hours for the 25 EV scenario, compared to CTDE-DDPG's 5.64 and 8.25 hours, respectively. This demonstrates better scalability and efficiency, especially with larger EV fleets, as the termination criterion ensures faster and more optimized solutions.

Finally, with regard to the ability to meet the target SoC, the proposed method demonstrates excellent performance. For the 5 EV scenario, it achieves a minimum SoC of 0.91, a maximum SoC of 0.96, and a mean SoC of 0.93. In the 25 EV scenario, the mean SoC of the proposed method is highest among other methods. The method's flexibility ensures that, even with a larger number of EVs, the target SoC is consistently met, reflecting its ability to maintain user satisfaction with the system's charging behavior.

Table 3: Performance comparison across scenarios and models

Model	**Metric**	**5 EVs**	**25 EVs**
CTDE-DDPG	Cost (€/kWh)	0.0538	0.0428
	PV utilization (%)	83	100
	Violation count	1 SoC, 4 peak	23 SoC
	Computation Time (hr)	5.64	8.25
	[Min SoC, Max SoC, Mean SoC]	[0.84, 0.96, 0.91]	[0.73, 0.81, 0.77]
CTDE-SAC	Cost (€/kWh)	0.0518	0.0469
	PV utilization (%)	98	100
	Violation count	1 SoC, 2 peak	0
	Computation Time (hr)	0.60	0.88
	[Min SoC, Max SoC, Mean SoC]	[0.79, 0.90, 0.85]	[0.81, 0.91, 0.85]
CTDE-SAC with Flexibility	Cost (€/kWh)	0.0532	0.0481
	PV utilization (%)	94	100
	Violation count	3 peak	1 SoC
	Computation Time (hr)	0.53	0.78
	[Min SoC, Max SoC, Mean SoC]	[0.91, 0.96, 0.93]	[0.75, 0.98, 0.88]

Figure 1 compares the cumulative EV power profiles achieved by different methods throughout the day, taking into account the load, PV generation, and electricity cost. As shown in the figure, CTDE-SAC Flexibility demonstrates the highest amount of discharging compared to other methods and prevents excessive charging peaks. It also avoids discharging in periods of time when the PV generation is high. These patterns show that the method adjusts charging and discharging based on what's happening in the system, making it more flexible in terms of responding to different system conditions.

A closer look at the EVs charging plan generated by the CTDE-SAC Flexibility method in Fig. 2 reveals that the EVs start charging soon after arrival, particularly during periods of high PV generation. Once they approach or reach their target SoC, the charging rate decreases, and by the end of the day, the EVs even discharge energy without dropping below their final SoC limits. This helps mitigate load imbal-

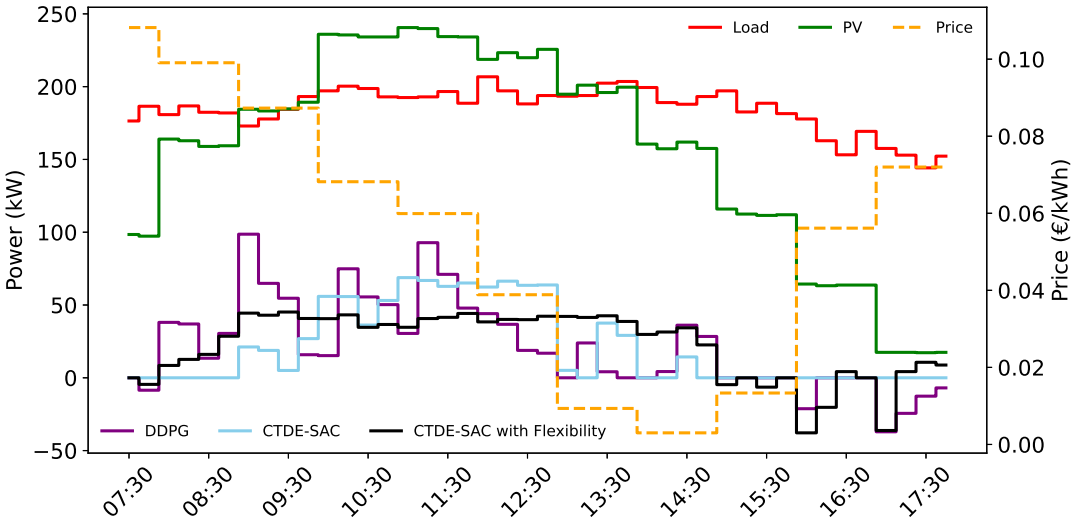

Fig. 1: Comparison of Load, PV, and Proposed Methods vs. Price

ances in the evening hours when PV generation is low and electricity prices rise.

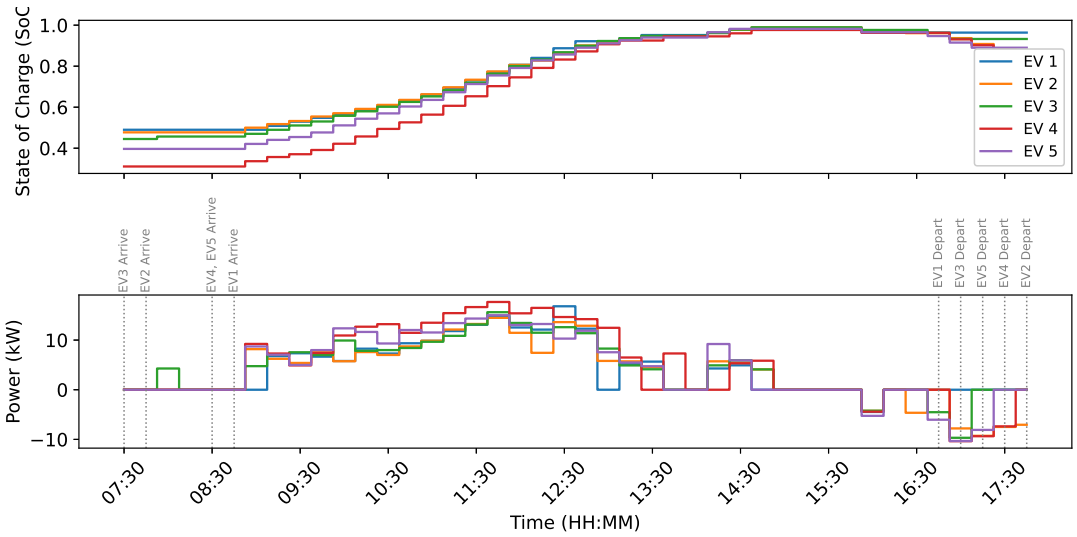

Fig. 2: Soc vs. Time for CTDE-SAC with flexibility

In Fig. 3, the flexibility characteristics of the different methods are illustrated. The plot on the left shows the standard deviation of cumulative EV power. CTDE-DDPG exhibits the highest variability, indicating frequent and significant shifts in power usage. Whereas, CTDE-SAC Flexibility demonstrates the lowest standard deviation, suggesting a more stable and consistent charging schedule. The right plot shows the average flexibility offered in kW by each method, which indicates the system's capacity to adjust power usage in response to changing grid conditions. While CTDE-DDPG achieves the highest average flexibility, it does so at the expense of stability. CTDE-SAC, although more stable, offers limited flexibility. CTDE-SAC Flexibility presents the highest average flexibility. This highlights its potential as a scalable and grid-friendly solution for dynamic EV scheduling scenarios.

Lastly, Fig. 4 shows the average reward vs. episode number for our considered methods. Both CTDE-SAC methods fulfill the termination criteria early on and achieve a high reward, while CTDE-DDPG shows fluctuation and a higher running time. This suggests that the CTDE-SAC methods provide faster convergence and more stable performance compared to CTDE-DDPG, which suffers from reward swings and high computation time.

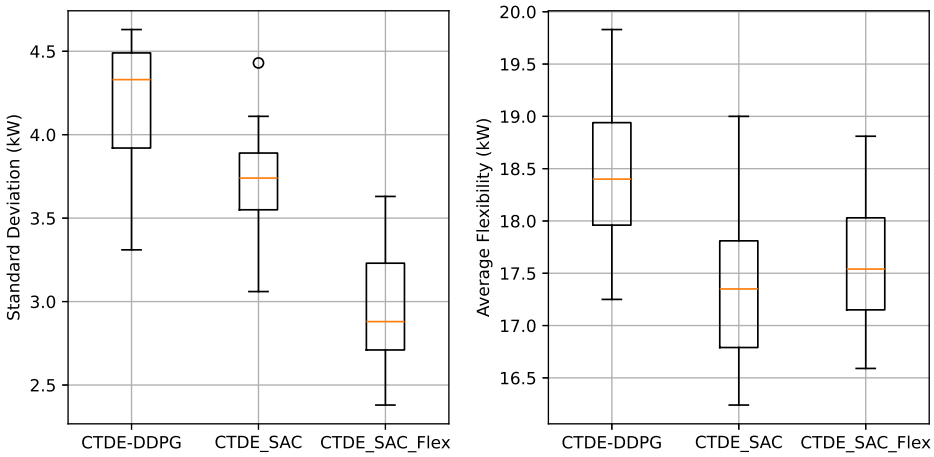

Fig. 3: Comparison of flexibility offered by different methods

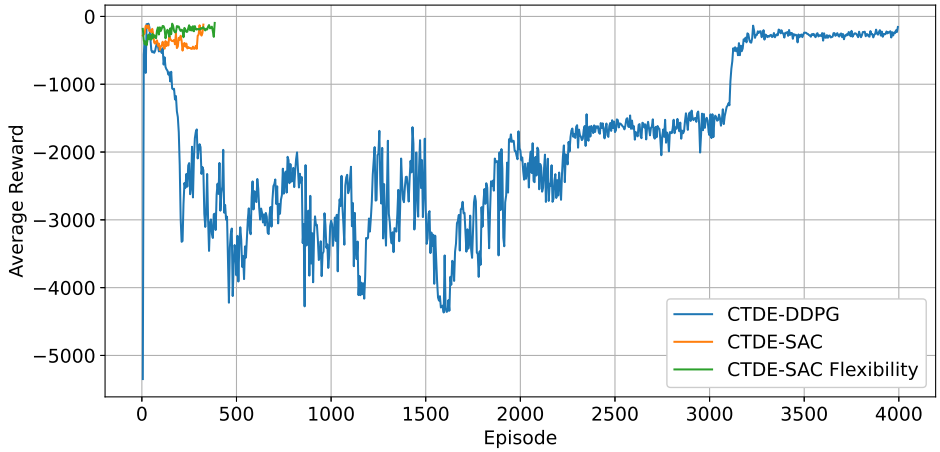

Fig. 4: Reward vs. episode comparison

4 Conclusion

This paper proposed a smart charging framework for bidirectional EV scheduling based on SAC, combined with a CTDE framework to ensure scalability. Simulation results show that our approach achieves higher PV energy utilization and greater flexibility compared to state-of-the-art DDPG-based methods, while maintaining similar charging costs. Moreover, the CTDE framework significantly reduces computation time compared to fully centralized methods, which is crucial for enabling real-time control and large-scale fleet management. By preserving dynamic flexibility and improving renewable energy integration, our proposed method offers an effective and scalable solution for future energy systems with high EV penetration. Future work will focus on implementing the proposed framework in real-world testbeds and exploring its performance under realistic operational uncertainties.

Acknowledgments

This work was supported by the German Research Foundation as part of Germany's Excellence Strategy—EXC 2050/1—Cluster of Excellence "Centre for Tactile Internet with Human-in-the-Loop" (CeTI) of Technische Universität Dresden under Project ID 390696704 and the research project DymoBat (Ref.: 03EI6082A) funded by the Economic Affairs and Climate Action (BMWK),

References

[1] *Global EV Outlook 2023 - IEA, Paris*, https://www.iea.org/reports/global-ev-outlook-2023, accessed on 2024-10-30.

[2] *Renewables 2023, Analysis and forecasts to 2028 - IEA, Paris*, https://www.iea.org/reports/renewables-2023, accessed on 2024-10-30.

[3] S. Shen, R. Habeeb, N. Alirezaei, K. Schert, R. Lehnert and F. H. P. Fitzek, *A Heuristic for Bi-directional Charging of Fleet EVs*, 2024 IEEE Sustainable Power and Energy Conference (iSPEC), Kuching, Sarawak, Malaysia, 2024, pp. 595–601, doi: 10.1109/iSPEC59716.2024.10892435.

[4] *Frequency-Containment-Reserve*, https://www.regelleistung.net/de-de/Allgemeine-Infos/Arten-der-Regelreserve/Frequency-Containment-Reserve, accessed on 2024-10-30.

[5] *Redispatch 3.0*, https://www.transnetbw.de/de/unternehmen/politik-und-regulierung/ konzepte/Redispatch-30, accessed on 2024-10-30.

[6] Y. Li, M. Han, Z. Yang and G. Li, *Coordinating Flexible Demand Response and Renewable Uncertainties for Scheduling of Community Integrated Energy Systems With an Electric Vehicle Charging Station: A Bi-Level Approach*, IEEE Transactions on Sustainable Energy, vol. 12, no. 4, pp. 2321–2331, Oct. 2021, doi: 10.1109/TSTE.2021.3090463.

[7] H. M. Abdullah, A. Gastli and L. Ben-Brahim, *Reinforcement Learning Based EV Charging Management Systems–A Review*, IEEE Access, vol. 9, pp. 41506–41531, 2021, doi: 10.1109/ACCESS.2021.3064354.

[8] Y. Wen, P. Fan, J. Hu, S. Ke, F. Wu and X. Zhu, *An Optimal Scheduling Strategy of a Microgrid with V2G Based on Deep Q-Learning*, Sustainability, vol. 14, no. 16, p. 10351, 2022, doi: 10.3390/su141610351.

[9] S. J. Sultanuddin, R. Vibin, A. R. Kumar, N. R. Behera, M. J. Pasha and K. K. Baseer, *Development of Improved Reinforcement Learning Smart Charging Strategy for Electric Vehicle Fleet*, Journal of Energy Storage, vol. 64, p. 106987, 2023, doi: 10.1016/j.est.2023.106987.

[10] H. Li, X. Dai, S. Goldrick, R. Kotter, N. Aslam and S. Ali, *Reinforcement Learning for EV Fleet Smart Charging with On-Site Renewable Energy Sources*, Energies, vol. 17, no. 21, p. 5442, 2024, doi: 10.3390/en17215442.

[11] M. Sharif and H. Seker, *Smart EV Charging With Context-Awareness: Enhancing Resource Utilization via Deep Reinforcement Learning*, IEEE Access, vol. 12, pp. 7009–7027, 2024, doi: 10.1109/ACCESS.2024.3351360.

[12] E. Aydin and S. Iqbal, *Reinforcement Learning-Based Optimization for Electric Vehicle Dispatch in Renewable Energy Integrated Power Systems*, in 2024 IEEE Energy Conversion Congress and Exposition (ECCE), Phoenix, AZ, USA: IEEE, Oct. 2024, pp. 1297–1304, doi: 10.1109/ECCE55643.2024.10861099.

[13] A. Shojaeighadikolaei, Z. Talata, and M. Hashemi, *Centralized vs. Decentralized Multi-Agent Reinforcement Learning for Enhanced Control of Electric Vehicle Charging Networks*, arXiv:2404.12520, Apr. 18, 2024, doi: 10.48550/arXiv.2404.12520.

[14] F. Liu et al., *Reinforcement Learning-based Approach for Vehicle-to-Building Charging with Heterogeneous Agents and Long Term Rewards*, arXiv preprint arXiv:2502.18526, Feb. 2025, doi: 10.48550/arXiv.2502.18526.

[15] F. Zhang et.al., *CDDPG: A Deep-Reinforcement-Learning-Based Approach for Electric Vehicle Charging Control*, IEEE Internet Things J., vol. 8, no. 5, pp. 3075–3087, Mar. 2021, doi: 10.1109/JIOT.2020.3015204.

[16] F. Tuchnitz et.al., *Development and Evaluation of a Smart Charging Strategy for an Electric Vehicle Fleet Based on Reinforcement Learning*, Applied Energy, vol. 285, p. 116382, Mar. 2021, doi: 10.1016/j.apenergy.2020.116382.

[17] T. Haarnoja et al., *Soft Actor-Critic Algorithms and Applications*,*arXiv preprint arXiv:1812.05905*, Jan. 29, 2019.

[18] Y. Chen, X. Zhang, X. Wang, Z. Xu, X. Shen, and W. Zhang, *Rethinking Soft Actor-Critic in High-Dimensional Action Spaces: The Cost of Ignoring Distribution Shift*, arXiv preprint arXiv:2410.16739, Apr. 2025, doi: 10.48550/arXiv.2410.16739.

Presenter Biography

Shiwei Shen holds a B.Sc. and an M.Sc. in Electrical Engineering from RWTH Aachen University. He is currently a junior researcher at the Deutsche Telekom Chair of Communication Networks, TU Dresden. His current work focuses on the DymoBat project, which aims to develop marketable solutions for future energy grid management by utilizing distributed energy resources and applying 5G technologies.