2026 IEEE 46th International Conference on Distributed Computing Systems (ICDCS) | 979-8-3195-2979-4/26/$31.00 ©2026 IEEE | DOI: 10.1109/2575-8411.2026.00094

A Cyber-Physical Cooperative Scheme for Energy Scheduling and Dynamic Pricing in Charging Station Alliances via Cloud-Edge Intelligence

1st Bing Lin
College of Physics and Energy, Fujian Normal University
Fuzhou, China
WheelLX@163.com

2nd Kai He
College of Physics and Energy, Fujian Normal University
Fuzhou, China
kaihehk@163.com

3rd Xingyuan Lin
School of Computer Science and Engineering, Macau University of Science and Technology
Macau, China
suoluneye@163.com

4th Jingtao Chen
College of Computer and Cyber Security, Fujian Normal University
Fuzhou, China
cxchen721@163.com

5th Huaming Wu*
Center for Applied Mathematics, Tianjin University
Tianjin, China
whming@tju.edu.cn

***Abstract*—Coordinating geographically dispersed Photovoltaic-Energy Storage Charging Stations (PSCSs) constitutes a complex Cyber-Physical System (CPS) characterized by the tight coupling of physical energy flows and cyber information exchanges. However, the efficient operation of this CPS is challenged by the stochasticity of Electric Vehicle (EV) demand, the intermittency of renewable energy, and the inherent tendency toward monopoly pricing in fully cooperative settings. To address these challenges, this paper proposes a cyber-physical cooperative control scheme for the Charging Station Alliance (CSA) utilizing a Cloud-Edge collaborative framework based on Centralized Training and Distributed Execution (CTDE). First, a Markov game model is formulated in which PSCSs operate as fully cooperative Internet of Things (IoT) edge agents. To curb irrational monopoly pricing behaviors, an Inverse Demand Function (IDF) is integrated with a gravity-based attraction model to explicitly quantify EV user sensitivity to charging prices and spatial distance. Second, a Cosine Annealing-based Multi-Agent Twin Delayed Deep Deterministic Policy Gradient (CA-MATD3) algorithm is proposed. This architecture enables lightweight edge inference to overcome communication latency constraints while leveraging cloud resources for global policy optimization. By dynamically adjusting learning rates, CA-MATD3 effectively mitigates non-stationarity in dynamic IoT environments. Extensive simulation results demonstrate that the proposed scheme prevents excessive price hikes, and CA-MATD3 outperforms benchmark algorithms by 1.3%–38.6% in terms of overall profit under various scenarios.**

***Index Terms*—Cyber-Physical Systems (CPS), Internet of Things (IoT), Charging Station Alliance (CSA), Centralized Training and Distributed Execution (CTDE), Multi-Agent Deep Reinforcement Learning (MADRL).**

This work is supported by the Fujian Provincial Special Fund Project for Promoting High-Quality Development of Marine and Fishery Industries under Grant No. FJHYF-L-2025-07-004, the University-Industry Cooperation of Fujian Province under Grant No. 2022H6024, the Tianjin Natural Science Foundation Project under Grant No. 25JCYBJC01540, and the Tianjin Municipal Science and Technology Major Program under Grant No. 25ZXZSSS00390. (*Corresponding author: Huaming Wu*)

I. Introduction

With the escalating global energy crisis and rising carbon emissions, energy strategies have increasingly pivoted towards sustainable systems driven by advanced information technologies [1], [2]. Electric Vehicles (EVs), acting as mobile loads and storage units, are pivotal components of these systems [3]. According to the International Energy Agency, the global stock of EVs is projected to surge to 525 million by 2035 [4]. The integration of such a massive scale of EVs into the power grid constitutes a complex Cyber-Physical System (CPS), where the physical energy flows are tightly coupled with cyber control signals. However, the stochastic and bursty nature of EV charging behaviors exacerbates peak-to-valley load differences [5], [6], imposing significant strain on this coupled infrastructure.

Photovoltaic-Energy Storage Charging Stations (PSCSs), integrated with Photovoltaic (PV) system and Energy Storage System (ESS), serve as critical Internet of Things (IoT) edge nodes to mitigate these impacts. By performing peak-shaving and valley-filling locally [7], PSCSs can alleviate grid pressure [8]. Nevertheless, operating these distributed entities efficiently is challenging due to the inherent uncertainties in user demand and renewable generation. These uncertainties often lead to insufficient profitability and system instability. Consequently, developing effective cyber-physical control schemes that combine energy scheduling [9] and dynamic pricing [10] has become imperative.

To realize these schemes, existing control architectures generally fall into two categories: centralized and distributed. Centralized cloud-based approaches [11]–[13] aggregate global data for optimization. However, in massive IoT networks, transmitting real-time sensor data from distributed stations to a central cloud incurs high bandwidth costs and significant communication latency, which is effectively mismatched with the real-time control requirements of CPS [14]. Conversely, fully distributed approaches [15], [16] allow local edge agents to make decisions based solely on local observations. While this ensures responsiveness, the lack of global coordination often causes agents to fall into local optima [13], leading to inefficient resource utilization across the alliance.

To bridge the gap between global coordination and edge responsiveness, the Cloud-Edge collaborative framework based on Centralized Training and Distributed Execution (CTDE) [17] has emerged as a promising solution for AI-driven IoT systems [18], [19]. Multi-Agent Deep Reinforcement Learning (MADRL) adopting the CTDE framework enables the cloud layer to facilitate optimized policy learning using global data, while the edge layer executes lightweight policies with millisecond-level latency. In this context, we introduce the Charging Station Alliance (CSA) [20], [21], a cyber-physical aggregation framework designed to coordinate physically distributed entities. By adopting this Cloud-Edge architecture, the CSA maximizes overall profitability while maintaining the autonomy of distributed execution essential for IoT resilience.

The aforementioned works provide inspiration for energy scheduling and pricing in PSCSs. However, considering the CPS characteristics, they suffer from the following limitations:

- Existing works often model PSCSs as competitive entities. In contrast, within a unified management framework like a CSA, PSCSs operate as fully cooperative IoT edge agents to maximize the overall profit of the alliance.
- In a fully cooperative setting, autonomous agents may collectively overprice to maximize profits, resulting in monopoly pricing [22]. This behavior arises because traditional algorithms often lack explicit modeling of the cyber-physical coupling between user demand elasticity and pricing signals. Such monopolistic practices are inconsistent with realistic market principles and can lead to user attrition. Existing research has not effectively addressed this issue.
- Existing DRL-based methods often suffer from low convergence efficiency and poor global search capability in dynamic IoT environments. Single-optimizer methods (e.g., Adam, SGD) [23] struggle to adapt to the non-stationarity caused by the complex interactions among multiple heterogeneous agents and environmental uncertainties.

To address these limitations, this paper proposes a cyber-physical cooperative scheme for energy scheduling and dynamic pricing. The underlying algorithmic architecture, designed to balance global optimization with real-time edge execution, is shown in Fig. 1. Specifically, the main contributions of this paper are:

- A Markov game model is formulated, where PSCSs are considered as fully cooperative IoT edge agents, while EVs and the grid constitute the physical environment. Agents interact with the environment to perform joint energy scheduling and dynamic pricing based on PV generation, EV energy demand, ESS state of charge (SOC), and Time-of-Use (TOU) price of the grid.
- The Inverse Demand Function (IDF) is employed to model the sensitivity of EVs to charging prices. It establishes an explicit inverse relationship between EV energy demand and the charging price. Based on the IDF, a universal gravitation model is utilized to formulate the attraction of each PSCS to EVs. By incorporating user sensitivity, the IDF mechanism inherently suppresses irrational monopoly pricing behaviors, aligning the system's cyber decisions with market sustainability.
- A Cosine Annealing-based Multi-Agent Twin Delayed Deep Deterministic Policy Gradient (CA-MATD3) algorithm is proposed. This algorithm integrates a cosine annealing method to dynamically adjust learning rates, thereby mitigating the non-stationarity inherent in multi-agent environments. Simulation results show that CA-MATD3 significantly improves convergence efficiency and empowers resource-constrained edge nodes to achieve near-optimal global performance.

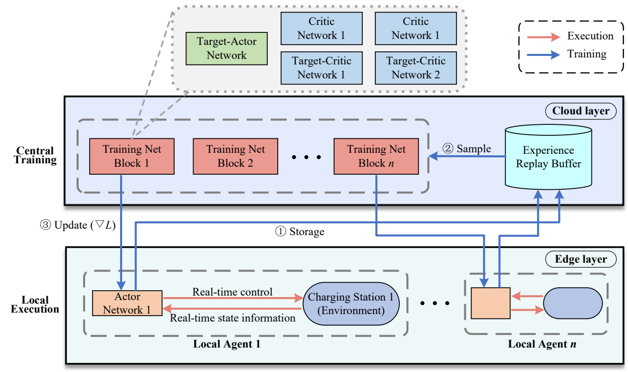


Fig. 1: Schematic diagram of the proposed CA-MATD3 learning architecture based on the Cloud-Edge collaborative CTDE framework.

The rest of this article is organized as follows. Section II presents the system model of the cyber-physical service for the CSA. In Section III, the details of the proposed CA-MATD3 are described. Section IV evaluates the performance of CA-MATD3 compared to benchmark algorithms.

II. System Model and Definitions

In this section, we formulate the problem of the energy scheduling and dynamic pricing service for the CSA. From the perspective of CPS, the CSA is modeled as a hierarchical system comprising a Physical Layer and a Cyber Layer, as illustrated in Fig. 2. The Physical Layer encompasses the power grid, m EVs, and the hardware infrastructure of the CSA, which integrates n geographically dispersed PSCSs. This layer handles the physical energy flows and storage. The Cyber Layer consists of distributed IoT sensors, Intelligent Edge Controllers (IECs) deployed at each PSCS, and a central

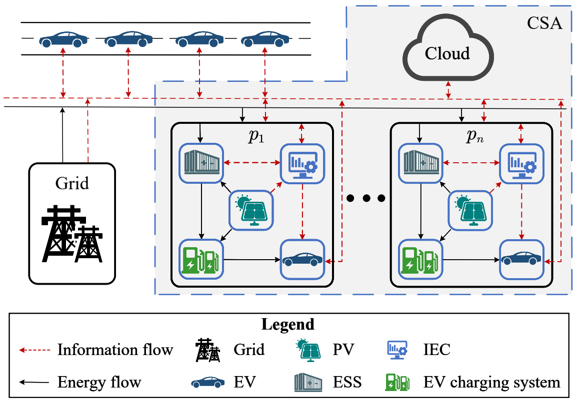


Fig. 2: Architecture of the Cyber-Physical Energy System for CSA involving Cloud-Edge collaboration.

cloud server. This layer handles data sensing, computation, and control command transmission.

Although these PSCSs may belong to different operators, they operate as fully cooperative IoT nodes within the CSA to maximize the overall profit through cloud-edge collaboration.

A. Physical Nodes: PSCSs with Edge Intelligence

There are n PSCSs in the CSA, denoted as $\boldsymbol{P} = \{p_1, p_2, \ldots, p_n\}$. Each PSCS p_i is a cyber-physical node equipped with a PV system, an ESS, an EV charging system, and an IEC.

To reduce PV curtailment [24], the IEC prioritizes directing PV output energy to the EV charging system, with surplus energy stored in the ESS. Therefore, at time slot t, the PV energy accommodated by p_i, $q_{i,t}^{\text{pv}}$, is expressed as Eq. (1).

$$q_{i,t}^{\text{pv}} = \begin{cases} \min\left(q_{i,t}^{\text{pv,out}}, q_{i,t}^{\text{ev}} - q_{i,t}^{\text{ess}}\right), & q_{i,t}^{\text{ess}} < 0, \\ \min\left(q_{i,t}^{\text{pv,out}}, q_{i,t}^{\text{ev}}\right), & q_{i,t}^{\text{ess}} \geqslant 0, \end{cases} \tag{1}$$

where $q_{i,t}^{\text{pv,out}}$ is the real-time PV generation sensed by local IoT sensors, and $q_{i,t}^{\text{ev}}$ is the actual charging energy delivered to EVs. $q_{i,t}^{\text{ess}}$ represents the ESS charging/discharging command executed by the IEC, where $q_{i,t}^{\text{ess}} < 0$ indicates charging and $q_{i,t}^{\text{ess}} \geqslant 0$ indicates discharging.

The ESS acts as a physical energy buffer to smooth fluctuations. Its operation is constrained by physical limits monitored by the IEC. p_i's ESS energy flow $q_{i,t}^{\text{ess}}$ is calculated by Eq. (2).

$$q_{i,t}^{\text{ess}} = \left(soc_{i,t}^{\text{ess}} - soc_{i,t+1}^{\text{ess}}\right) \cdot \frac{C_i^{\text{ess}}}{\zeta}, \tag{2}$$

where C_i^{ess} denotes the ESS capacity. $soc_{i,t}^{\text{ess}}$ denotes the State of Charge (SOC) sensed at time slot t, updated by the physical dynamics in Eq. (3).

$$soc_{i,t+1}^{\text{ess}} = \begin{cases} soc_{i,\max}^{\text{ess}}, & soc_{i,t}^{\text{ess}} - \frac{q_{i,t}^{\text{ess}} \cdot \zeta}{C_i^{\text{ess}}} \geqslant soc_{i,\max}^{\text{ess}}, \\ soc_{i,t}^{\text{ess}} - \frac{q_{i,t}^{\text{ess}} \cdot \zeta}{C_i^{\text{ess}}}, & soc_{i,\min}^{\text{ess}} < soc_{i,t}^{\text{ess}} - \frac{q_{i,t}^{\text{ess}} \cdot \zeta}{C_i^{\text{ess}}} < soc_{i,\max}^{\text{ess}}, \\ soc_{i,\min}^{\text{ess}}, & soc_{i,t}^{\text{ess}} - \frac{q_{i,t}^{\text{ess}} \cdot \zeta}{C_i^{\text{ess}}} \leqslant soc_{i,\min}^{\text{ess}}, \end{cases} \tag{3}$$

where $soc_{i,\max}^{\text{ess}}$ and $soc_{i,\min}^{\text{ess}}$ are the physical safety bounds. ζ is the efficiency coefficient defined as Eq. (4).

$$\zeta = \begin{cases} \eta^c, & q_{i,t}^{\text{ess}} < 0, \\ \frac{1}{\eta^d}, & q_{i,t}^{\text{ess}} \geqslant 0, \end{cases} \tag{4}$$

where $\eta^c, \eta^d \in [0, 1]$ are charging/discharging efficiencies [25].

To account for hardware aging, the ESS degradation cost $c_{i,t}^{\text{ess}}$ is defined as Eq. (5).

$$c_{i,t}^{\text{ess}} = \zeta \cdot \beta \cdot \left|q_{i,t}^{\text{ess}}\right|, \tag{5}$$

where β is the degradation coefficient.

The EV charging system facilitates energy transfer. The actual energy delivered, $q_{i,t}^{\text{ev}}$, is physically constrained by the available supply from PV, ESS, and the grid, as expressed in Eq. (6).

$$q_{i,t}^{\text{ev}} = \min\left(q_{i,t}^{\text{pv}} + q_{i,t}^{\text{ess}} + q_{i,t}^{\text{grid}}, q_{i,t}^{\text{ev,tot}}\right), \tag{6}$$

where $q_{i,t}^{\text{grid}}$ is the energy purchased from the grid via the smart meter, and $q_{i,t}^{\text{ev,tot}}$ is the aggregated demand. The revenue $u_{i,t}$ is defined as Eq. (7).

$$u_{i,t} = q_{i,t}^{\text{ev}} \cdot \omega_{i,t}^{\text{ev}}, \tag{7}$$

where $\omega_{i,t}^{\text{ev}}$ is the dynamic pricing signal broadcasted by p_i.

The IEC collects local IoT sensor data (i.e., PV output and SOC) and uploads it to the cloud. Simultaneously, it executes local control commands (i.e., pricing, ESS scheduling, and grid purchasing) derived from the edge policy or cloud instructions.

B. Mobile Users: EVs with Cyber-Physical Coupling

There are m EVs, $\boldsymbol{E} = \{e_1, e_2, \ldots, e_m\}$. We model EVs as rational mobile agents that interact with the CPS based on physical distance and cyber pricing signals. This rationality implies a negative correlation between an EV's target State of Energy (SOE) and the charging price. This inverse demand relationship [26] acts as a cyber-physical coupling mechanism that discourages PSCSs from monopoly pricing. We adopt the IDF to formulate e_j's target SOE at p_i, $q_{j,i,t}^{\text{ev,tar}}$, as Eq. (8).

$$q_{j,i,t}^{\text{ev,tar}} = C_j^{\text{ev}} \cdot e^{\frac{-\left(\omega_{i,t}^{\text{ev}}\right)^2}{2a^2}}, \tag{8}$$

where C_j^{ev} is the battery capacity and a is the inverse demand coefficient. e_j's actual demand $q_{j,i,t}^{\text{ev,dem}}$ accounts for travel consumption, as described in Eq. (9).

$$q_{j,i,t}^{\text{ev,dem}} = q_{j,i,t}^{\text{ev,tar}} - \left(q_{j,t}^{\text{ev}} - l_j \cdot d_{i,j,t}\right), \tag{9}$$

where $q^{\text{ev}}_{j,t}$ is the current SOE, l_j is energy consumption per km, and $d_{i,j,t}$ is the physical distance.

Therefore, e_j's charging cost at p_i at time slot t, $c^{\text{ev}}_{j,i,t}$, is defined as Eq. (10).

$$c^{\text{ev}}_{j,i,t} = \omega^{\text{ev}}_{i,t} \cdot q^{\text{ev,dem}}_{j,i,t}. \tag{10}$$

To model the spatial-temporal selection behavior of EVs, we employ a universal gravitation model [27], which quantifies the attraction $F_{i,j,t}$ of a cyber-physical node p_i to a mobile user e_j, as defined in Eq. (11).

$$F_{i,j,t} = \frac{1}{c^{\text{ev}}_{j,i,t}} \cdot \frac{q^{\text{ev,dem}}_{j,i,t} \cdot (C^{ev}_j - q^{ev}_{j,t})}{(d_{i,j,t})^2}, \tag{11}$$

where $\frac{1}{c^{\text{ev}}_{j,i,t}}$ corresponds to the gravitational constant, $q^{\text{ev,dem}}_{j,i,t}$ corresponds to the mass of the central body, $C^{ev}_j - q^{ev}_{j,t}$ corresponds to the masses of the planet, and $d_{i,j,t}$ corresponds to the distance between celestial bodies.

Each rational EV e_j selects the most attractive PSCS node $p_{\varrho_{e_j}}$ for charging, where the decision ϱ_{e_j} is expressed as Eq. (12).

$$\varrho_{e_j} = \arg\max_i (F_{i,j,t}). \tag{12}$$

Therefore, the aggregated demand $q^{\text{ev,tot}}_{i,t}$ sensed by p_i is described as Eq. (13).

$$q^{\text{ev,tot}}_{i,t} = \sum_{j=1}^{m} x_{i,j} \cdot q^{\text{ev,dem}}_{j,i,t}, \tag{13}$$

$$x_{i,j} = \begin{cases} 1, & \text{if } \varrho_{e_j} = i, \\ 0, & \text{else}, \end{cases} \tag{14}$$

where $x_{i,j}$ denotes e_j's selecting decision. When $x_{i,j} = 1$, it means p_i is selected by e_j for charging. When $x_{i,j} = 0$, it means the opposite.

C. Cloud-Edge Collaborative Architecture

As illustrated in Fig. 1, the proposed framework adopts a Cloud-Edge collaborative architecture based on the CTDE paradigm.

1) Edge Layer (Distributed Execution): Each PSCS operates as an autonomous edge agent. The local IEC hosts an actor network responsible for making real-time actuation commands (i.e., pricing $\omega^{\text{ev}}_{i,t}$, ESS scheduling $q^{\text{ess}}_{i,t}$, and grid purchasing $q^{\text{grid}}_{i,t}$) based on local IoT observations. This edge-based execution minimizes the latency caused by transmitting high-frequency control signals over the network.

The dynamic pricing signal $\omega^{\text{ev}}_{i,t}$ is updated locally to shape user demand, as shown in Eq. (15).

$$\omega^{\text{ev}}_{i,t+1} = \omega^{\text{ev}}_{i,t} + \lambda_{i,t}, \tag{15}$$

where $\lambda_{i,t}$ is the price fluctuation action determined by the edge policy.

The ESS and grid energy flows are controlled to satisfy the physical constraints. Specifically, the grid purchase command $q^{\text{grid}}_{i,t}$ is executed as Eq. (16).

$$q^{\text{grid}}_{i,t} = \max\left(0, q^{\text{ev,tot}}_{i,t} - q^{\text{pv}}_{i,t} - q^{\text{ess}}_{i,t}\right). \tag{16}$$

2) Cloud Layer (Centralized Training): While edge agents execute decisions distributedly, the cloud server functions as a centralized trainer. It aggregates historical data from all edge nodes to train the global policy, aiming to maximize the alliance's overall profit.

The cloud monitors the global grid interconnection status. The total energy purchased q^{grid}_t is constrained by the grid's capacity $q^{\text{grid}}_{\max}$, calculated as Eq. (17).

$$q^{\text{grid}}_t = \min\left(\sum_{i=1}^{n} q^{\text{grid}}_{i,t}, q^{\text{grid}}_{\max}\right), \tag{17}$$

where $q^{\text{grid}}_{\max}$ is the maximum deliverable energy of the grid. Therefore, the total energy cost c^{grid}_t is defined as Eq. (18).

$$c^{\text{grid}}_t = q^{\text{grid}}_t \cdot \omega^{\text{grid}}_t, \tag{18}$$

where ω^{grid}_t represents the TOU price of the grid at time slot t.

To achieve the optimal cooperative policy, the cloud formulates the global optimization problem $\mathcal{P}$ as Eq. (19).

$$\begin{aligned} \mathcal{P}:\ \max \quad & r_t = \sum_{i=1}^{n} \left(u_{i,t} - c^{\text{ess}}_{i,t}\right) - c^{\text{grid}}_t, \\ \text{s.t.} \quad & (1), (4)-(7). \end{aligned} \tag{19}$$

Upon training completion, the optimized neural network parameters are downloaded from the cloud to the edge agents. This updates the local inference models, enabling the IECs to generate improved control strategies autonomously during the subsequent distributed execution phase.

III. OUR PROPOSED CA-MATD3

In this section, we first introduce the Markov game model for our proposed cyber-physical cooperative scheme. Then, the CA-MATD3 algorithm is proposed for solving the energy scheduling and dynamic pricing problem of CSA within the Cloud-Edge architecture.

A. Markov Game Model

Since our work focuses on the optimization problem in a fully cooperative multi-agent system, the Markov game model is formulated as a tuple $\langle n, \boldsymbol{S}, \{\boldsymbol{O}^i\}_{p_i \in \boldsymbol{P}}, \{\boldsymbol{A}^i\}_{p_i \in \boldsymbol{P}}, f, \{\boldsymbol{r}^i\}_{p_i \in \boldsymbol{P}}\rangle$. The definitions of each element are adapted to the CPS context as follows.

1) Agents (n): Each PSCS p_i acts as an autonomous IoT edge agent, totaling n agents.

2) State Space ($\boldsymbol{S}$): The global state of the cyber-physical environment at time slot t is denoted as $\boldsymbol{s}_t = \left\{t, \omega^{\text{grid}}_t, \boldsymbol{H}^{\text{ev}}_t, \boldsymbol{V}^{\text{soc}}_t, \boldsymbol{Q}^{\text{PV}}_t\right\}$, where $\boldsymbol{H}^{\text{ev}}_t = \left\{q^{\text{ev}}_{1,t}, \ldots, q^{\text{ev}}_{n,t}\right\}$ is the set of EVs' actual charging energy, $\boldsymbol{V}^{\text{soc}}_t =$

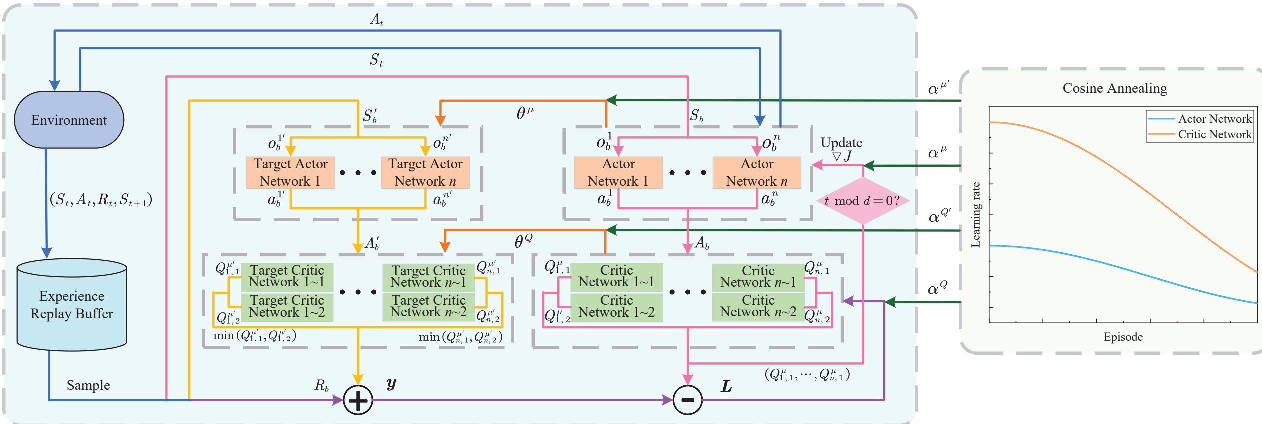


Fig. 3: Structure of the proposed CA-MATD3 algorithm within the Cloud-Edge architecture.

$\left\{soc^{\text{ess}}_{1,t}, \ldots, soc^{\text{ess}}_{n,t}\right\}$ is the set of ESS SOCs, and $\boldsymbol{Q}^{\text{pv}}_t = \left\{q^{\text{pv}}_{1,t}, \ldots, q^{\text{pv}}_{n,t}\right\}$ is the set of accommodated PV energy.

3) Observation Space ($\{\boldsymbol{O}^i\}$): Under partial observability constraints of edge nodes, the local sensing vector of agent p_i at time slot t is denoted as $\boldsymbol{o}^i_t = \left\{t, \omega^{\text{grid}}_t, q^{\text{ev}}_{i,t}, soc^{\text{ess}}_{i,t}, q^{\text{pv}}_{i,t}\right\}$.

4) Action Space ($\{\boldsymbol{A}^i\}$): Agent p_i's actuation command at time slot t, $\boldsymbol{a}^i_t = \{\lambda_{i,t}, q^{\text{ess}}_{i,t}\}$, controls the local price fluctuation and ESS charging/discharging.

5) State Transition (f): The function $f(\boldsymbol{s}_t, \boldsymbol{a}^1_t, \ldots, \boldsymbol{a}^n_t) \rightarrow \boldsymbol{s}_{t+1}$ describes the stochastic dynamics of the CPS driven by joint actions.

6) Reward Function ($\{\boldsymbol{r}^i\}$): The reward function corresponds to the objective defined in Eq. (19). For agent p_i, the reward r^i_t is defined as Eq. (20).

$$r^i_t = u_{i,t} - c^{\text{grid}}_{i,t} - c^{\text{ess}}_{i,t} - c^{\text{punish}}_{i,t} + r^{\text{energy}}_{i,t}, \tag{20}$$

$$c^{\text{punish}}_{i,t} = \begin{cases} \xi \cdot \left(soc^{\text{ess}}_{i,t} - soc^{\text{ess}}_{i,\max}\right) \cdot C^{\text{ess}}_i, & soc^{\text{ess}}_{i,t} > soc^{\text{ess}}_{i,\max}, \\ \xi \cdot \left(soc^{\text{ess}}_{i,\min} - soc^{\text{ess}}_{i,t}\right) \cdot C^{\text{ess}}_i, & soc^{\text{ess}}_{i,t} < soc^{\text{ess}}_{i,\min}, \end{cases} \tag{21}$$

$$r^{\text{energy}}_{i,t} = \psi \cdot q^{\text{ev}}_{i,t}, \tag{22}$$

where $c^{\text{grid}}_{i,t}$ is the energy cost of agent p_i at time slot t under the constraint in Eq. (17), and $c^{\text{grid}}_t = \sum_{i=1}^{n} c^{\text{grid}}_{i,t} \cdot c^{\text{punish}}_{i,t}$ is the ESS over-limit penalty of agent p_i at time slot t, and $r^{\text{energy}}_{i,t}$ is the guiding function of agent p_i at time slot t. ξ, ψ are the ESS over-limit penalty coefficient and the guiding coefficient, respectively.

B. CA-MATD3 Algorithm

As illustrated in Fig. 3, the proposed CA-MATD3 algorithm adopts the Cloud-Edge collaborative CTDE framework. Specifically, the architecture maintains four types of neural networks for the policy optimization of each edge agent: an actor network (deployed at the edge), two critic networks (residing in the cloud), and their corresponding target networks.

The actor network of agent p_i, μ_i, with parameter θ^{μ_i}, is deployed at the edge node (i.e., IEC) to generate real-time action policies. The action $\boldsymbol{a}^i_t$ is generated by Eq. (23).

$$\boldsymbol{a}^i_t = \mu_i\left(\boldsymbol{o}^i_t\right) + \mathcal{N}_{i,t}, \tag{23}$$

where $\mathcal{N}_{i,t}$ is the Gaussian noise of agent p_i, introduced to increase exploration.

The kth critic network of agent p_i, $Q^{\mu_i}_{i,k}$, with parameter $\theta^{Q_{i,k}}$, is maintained in the cloud to centrally evaluate the joint policies using global data. The input is a pair of global state-actions $(\boldsymbol{s}_t, \boldsymbol{a}_t)$, and the output is an action value function, $Q^{\mu_i}_{i,k}\left(\boldsymbol{s}_t, \boldsymbol{a}_t\right)$. For agent p_i, the actor network is updated by the policy gradient, $\nabla_{\theta^{\mu_i}} J\left(\mu_i\right)$, defined as Eq. (24).

$$\nabla_{\theta^{\mu_i}} J\left(\mu_i\right) \approx \frac{1}{B}\sum_{b=1}^{B} \nabla_{\theta^{\mu_i}}\mu_i\left(\boldsymbol{o}^i_b\right) \cdot \nabla_{\boldsymbol{a}^i} Q^{\mu_i}_{i,1}\left(\boldsymbol{s}_b, \boldsymbol{a}^1_b, \ldots, \boldsymbol{a}^n_b\right)|_{\boldsymbol{a}^i=\mu_i\left(\boldsymbol{o}^i_b\right)}, \tag{24}$$

where B is the batch size sampled from the experience replay buffer $\boldsymbol{D}$.

Using the sampled transitions, the centralized critic networks in the cloud are updated by gradient descent. The loss function, $L\left(\theta^{Q_{i,k}}\right)$, is defined as Eq. (25).

$$L\left(\theta^{Q_{i,k}}\right) = \frac{1}{B}\sum_{b=1}^{B}\left(y_i - Q^{\mu_i}_{i,k}\left(\boldsymbol{s}_b, \boldsymbol{a}^1_b, \ldots, \boldsymbol{a}^n_b\right)\right)^2, \forall k \in \{1, 2\}, \tag{25}$$

where y_i denotes the target value of agent p_i, defined as Eq. (26).

$$y_i = r^i_b + \gamma \min_{k=1,2} Q^{\theta^{Q'_{i,k}}}_{i,k}\left(\boldsymbol{s}'_b, \mu'_1(\boldsymbol{o}'^1_b) + \epsilon, \ldots, \mu'_n(\boldsymbol{o}'^n_b) + \epsilon\right), \tag{26}$$

where $Q^{\theta^{Q'_{i,k}}}_{i,k}$ and μ'_i denote the target critic and actor networks, respectively. $\epsilon \sim \text{clip}\left(\epsilon, -c, c\right)$ is a smoothing noise.

The target networks are updated in a soft manner to stabilize the training process, as shown in Eq. (27).

$$\begin{cases} \theta^{\mu'_i} = \tau \cdot \theta^{\mu_i} + (1 - \tau) \cdot \theta^{\mu'_i}, \\ \theta^{Q'_{i,k}} = \tau \cdot \theta^{Q_{i,k}} + (1 - \tau) \cdot \theta^{Q'_{i,k}}, \ \forall k \in \{1, 2\}, \end{cases} \tag{27}$$

where $\theta^{\mu'_i}$, $\theta^{Q'_{i,k}}$ are the parameters of the target actor network and the kth target critic network of agent p_i, respectively. $\tau \ll 1$ is the soft updating coefficient.

During training, an appropriate learning rate is crucial. In dynamic IoT environments, standard constant learning rates often lead to oscillation or premature convergence. To address this, we introduce the cosine annealing method [28]. This method allows the learning rate to gradually decay, promoting stable convergence to a better solution despite the non-stationarity of the CPS. The learning rate α is adjusted as Eq. (28).

$$\alpha = \alpha_{\min} + \frac{1}{2}\left(\alpha_{\max} - \alpha_{\min}\right)\left(1 + \cos\left(\frac{\rho}{\rho_{\max}} \cdot \pi\right)\right), \quad (28)$$

where $\alpha_{\min}$, $\alpha_{\max}$ are the minimum and maximum learning rates, respectively. ρ, $\rho_{\max}$ are the current episode number and the total number of episodes, respectively. π is the mathematical constant [29].

The pseudocode of the proposed CA-MATD3 algorithm is described in **Algorithm 1**. Initially, networks and the replay buffer $\boldsymbol{D}$ are instantiated (line 1). At each time slot, edge agents generate actions locally and store transitions (lines 5-9). Subsequently, the cloud updates critics (lines 11-13). Periodically, actor networks are updated and synchronized to edge nodes (lines 16-17).

Algorithm 1: CA-MATD3 Algorithm for Cloud-Edge Collaboration

Input: The total number of episodes $\rho_{\max}$, the total number of time slots T.
Output: The action policies of all edge agents.

1 **Initialization**: Network parameters, learning rates, and the experience replay buffer $\boldsymbol{D}$;
2 **for** $\rho = 1$ *to* $\rho_{\max}$ **do**
3   Receive the initial state $\boldsymbol{s}_1$;
4   **for** $t = 1$ *to* T **do**
5   **for** $i = 1$ *to* n **do**
6   Each edge agent p_i generates actuation command $\boldsymbol{a}_t^i$ using Eq. (23);
7   **end**
8   Execute joint actions $\boldsymbol{a}_t$, observe reward r_t and next state $\boldsymbol{s}_{t+1}$;
9   Store transition $(\boldsymbol{s}_t, \boldsymbol{a}_t, r_t, \boldsymbol{s}_{t+1})$ in $\boldsymbol{D}$;
10   **for** $i = 1$ *to* n **do**
11   Sample a random minibatch of B transitions from $\boldsymbol{D}$;
12   Calculate y_i using Eq. (26);
13   Update cloud critics $\theta^{Q_{i,1}}$, $\theta^{Q_{i,2}}$ using Eq. (25);
14   Adjust learning rates using Eq. (28);
15   **if** t mod $d = 0$ **then**
16   Update edge actor θ^{μ_i} using Eq. (24);
17   Update target networks using Eq. (27);
18   Adjust learning rates using Eq. (28);
19   **end**
20   **end**
21   **end**
22 **end**
23 **return** The action policies of all edge agents.

IV. Performance Evaluation

To evaluate the performance of the proposed CA-MATD3 and the effectiveness of the introduced IDF within the CPS, extensive experiments have been conducted in response to the following Research Questions (**RQs**).

RQ1: Does the IDF mechanism effectively model the cyber-physical coupling to facilitate rational pricing strategies, thereby avoiding monopoly pricing? (Section IV-C)

RQ2: Can CA-MATD3 adapt more efficiently to the dynamic IoT environment compared to other benchmark algorithms? (Section IV-D)

RQ3: Is CA-MATD3 better than other benchmark algorithms in terms of maximizing the overall profit of the CSA under different IoT scalability scenarios? (Section IV-E)

RQ4: Can CA-MATD3 generate precise actuation commands, enabling the edge nodes (i.e., PSCSs) to flexibly respond to the energy demand of EVs? (Section IV-F)

A. Basic Experimental Settings

All simulation experiments are run on the Win10 64-bit operating system with an Intel(R) Core(TM) i7-7700HQ CPU @ 2.80 GHz, 16 GB RAM, and an NVIDIA GeForce GTX 1060 graphics card. The proposed CA-MATD3 and other benchmark algorithms are all implemented and trained using Python 3.6 and PyTorch framework.

In our simulations, the dataset is constructed using the real IoT sensor data of EVs and PSCSs collected from companies in Fujian Province, China, which is structured into 24 time slots per day (i.e., T=24). 3 PSCSs with the same specifications (i.e., p_1, p_2, and p_3) and 120 EVs are selected to build the simulation environment. The parameters of PSCSs are shown in Table I [30]. The TOU price of the grid, the initial charging price, and the PV output energy of PSCSs are shown in Fig. 4. The PSCSs are distributed within a certain area of Fuzhou, Fujian Province, and the topology of the cyber-physical traffic network is shown in Fig. 5. The other experimental parameters are shown in Table II, and a total of 5000 training episodes are performed for all algorithms [31].

TABLE I: PSCS parameters

Parameter	Value	Parameter	Value
$\triangle t$	1 h	$q_{\max}^{\text{grid}}$	150 kWh
C^{ess}	250 kWh	β	0.08 CNY/kWh
soc_{init}	0.5	ξ	1500
$soc_{\max}$	0.97	ψ	8
$soc_{\min}$	0.03	η^d, η^c	0.95

B. Benchmark Algorithms

To analyze the performance of CA-MATD3 in maximizing the overall profit of the CSA within the Cloud-Edge architecture, three benchmark algorithms are introduced as

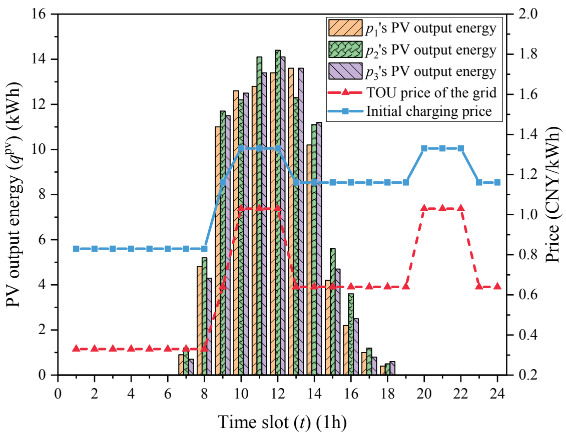


Fig. 4: The TOU price of the grid, the initial charging price, and the PV output energy sensed by IoT nodes.

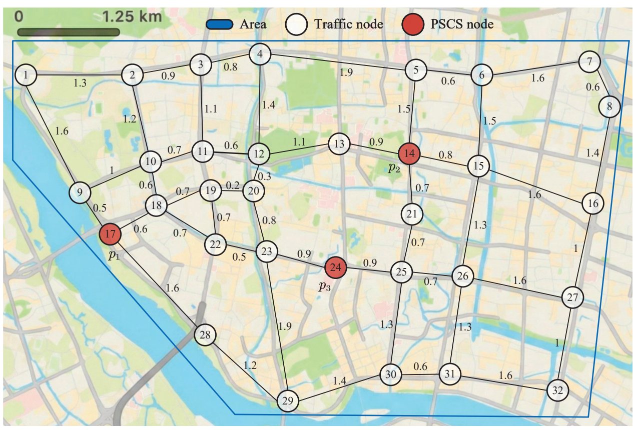


Fig. 5: Topology of the cyber-physical traffic network.

TABLE II: Experimental parameters

Parameters	Values
T	24
n	3
γ	0.935
τ	5×10^{-3}
D	5×10^{5}
B	2×10^{3}
ρ_{max}	5×10^{3}
λ_{max}	0.1
$\eta^{\mu}, \eta^{\mu'}$	1×10^{-4}
$\eta^{Q_1}, \eta^{Q_2}, \eta^{Q'_1}, \eta^{Q'_2}$	3×10^{-4}
d	2
ϵ	0.2
c	0.5

follows. The number of neural network layers for both the actor and critic networks in each benchmark algorithm, as well as experimental parameters, are the same as discussed above in CA-MATD3.

(**i**) Multi-Agent Deep Deterministic Policy Gradient (MADDPG) [32]: Extending DDPG to multi-agent domains, MADDPG assigns an individual actor-critic architecture to each agent. It adopts the CTDE framework, making it suitable for Cloud-Edge collaboration by utilizing global information during cloud training while relying solely on local observations for edge execution.

(**ii**) Multistep Multi-Agent Deep Deterministic Policy Gradient (MMADDPG) [33]: As an enhancement of MADDPG, MMADDPG incorporates a multistep reward mechanism. By prioritizing multistep returns over immediate rewards, it effectively balances long-term and short-term objectives, improving the learning efficiency of edge agents.

(**iii**) Multi-Agent Twin Delayed Deep Deterministic Policy Gradient (MATD3) [34]: MATD3 extends the Twin Delayed Deep Deterministic Policy Gradient (TD3) to multi-agent environments. By integrating double Q-learning, delayed policy updates, and target policy smoothing, it effectively mitigates Q-value overestimation, enhancing training stability in complex CPS scenarios.

C. Effectiveness of the IDF

To better reflect the cyber-physical coupling between the EVs' energy demand and the charging price, we conduct comparative experiments using different IDFs (i.e., the Gaussian function [35], the quadratic function [36], and the linear function [37]). The curves of these functions are shown in Fig. 6.

Fig. 7 displays the rewards of CA-MATD3 with different IDFs. Using the Gaussian function as the IDF, CA-MATD3 exhibits a steady increase in reward during training. Its reward stabilizes after about 3000 episodes and eventually converges at approximately 25000, significantly outperforming the other functions. This is because the Gaussian function better approximates realistic EV charging behavior by stabilizing energy demand variations at price extremes. In contrast, employing the quadratic or linear functions leads to either significant fluctuations or slower convergence rates, with both converging to a considerably lower reward of approximately 17000. These limitations arise from their failure to accurately capture the complex non-linear characteristics of the inverse demand relationship. Therefore, the Gaussian function is selected as the IDF for CA-MATD3 in the following experiments.

Fig. 8 displays the rewards of CA-MATD3 (IDF is Gaussian function) with different inverse demand coefficients (i.e., a). To determine a better inverse demand coefficient, we conduct a series of experiments with the coefficient increasing by 0.1. The experimental results indicate that when the coefficient is set to 1.5, CA-MATD3 achieves the fastest convergence and the highest converged reward. In contrast, when the coefficient is set to 1.4 or 1.6, both the convergence speed and the converged reward slightly decrease. Consequently, the inverse demand coefficient is fixed at 1.5 for CA-MATD3 in the following experiments.

Fig. 9 exhibits the charging price of each PSCS. To verify the effectiveness of the introduced IDF, we compare the charging price of each PSCS with and without the IDF (i.e., the Gaussian function). When the IDF is not introduced, the charging price of each PSCS is generally high and tends

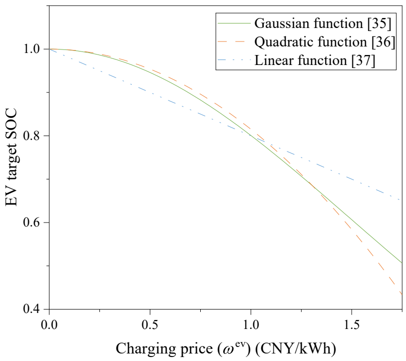


Fig. 6: Curves of different IDFs.

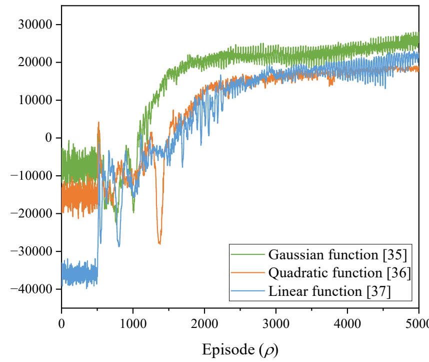


Fig. 7: Rewards of CA-MATD3 with different IDFs.

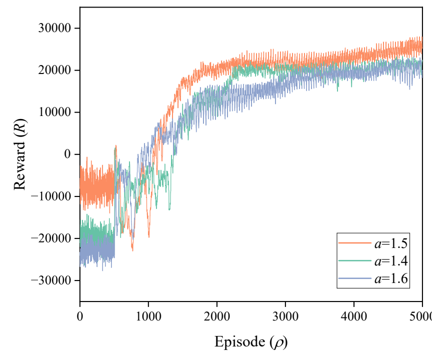
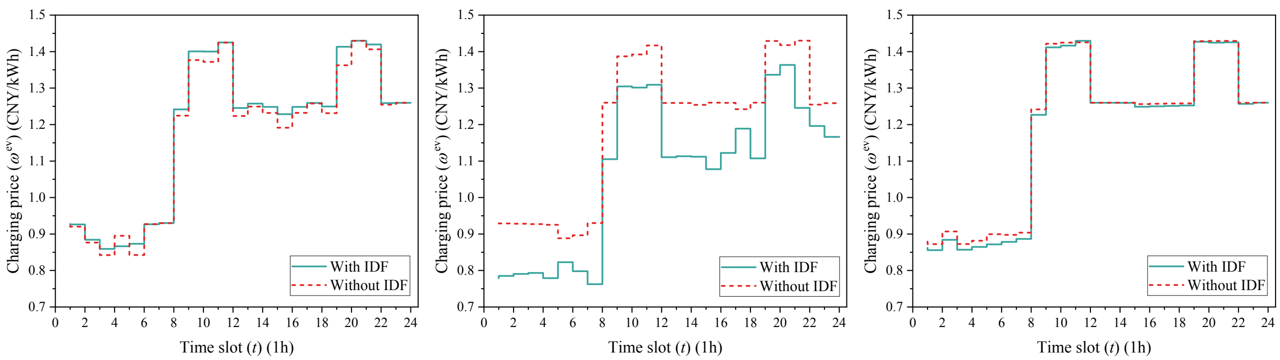


Fig. 8: Rewards of CA-MATD3 (IDF is Gaussian function) with different inverse demand coefficients (a).



(a) PSCS1 (p_1) (b) PSCS2 (p_2) (c) PSCS3 (p_3)

Fig. 9: Charging price of each PSCS.

to approach the maximum allowable changing price, which indicates that monopoly pricing exists in the fully cooperative pricing environment. When the IDF is introduced, the charging price of each PSCS universally reduces across all time slots. This is mainly because that the introduced IDF prevents the occurrence of monopoly pricing by effectively regulating the charging price.

After the introduction of the IDF, p_1's charging price slightly increases, which is shown in Fig. 9(a). This is mainly because p_1 is located relatively far from other PSCSs, a slight increase in charging price does not lead to a significant decrease in the energy demand of EVs. The increased unit profit for p_1 secures greater energy sales revenue. Fig. 9(b) displays that p_2's charging price decreases significantly. The main reason for this is that p_2 is located in the city center, where a large number of EVs require charging. A decrease in charging price significantly increases the energy demand of EVs, thereby improving the energy sales revenue. Fig. 9(c) indicates that p_3's charging price decreases slightly. This is because p_3 is slightly farther from the city center but close to hospitals and commercial areas. A slight reduction in changing price can increase the energy demand of nearby EVs, thereby maximizing the overall profit.

D. Convergence Evaluation

Fig. 10 displays the rewards of CA-MATD3 and other benchmark algorithms. In the first 500 episodes, the rewards of all algorithms fluctuate stably around their initial values. This is because these algorithms are offline learning methods, which require a certain number of transitions to be stored in the experience replay buffer before training begins. Therefore, in the early stage, all models randomly select actions to explore the environment and accumulate experience. After 500 episodes, their rewards all increase rapidly with significant fluctuations, which indicates that all models have started learning from the accumulated experience and are gradually converging toward better strategies.

The reward of CA-MATD3 gradually stabilizes around 2500 episodes and its value eventually converges at about 25000. While MATD3 achieves a converged reward of approximately 24000, it converges about 1000 episodes later than CA-MATD3. This is mainly because the introduced cosine annealing method can efficiently improve the training efficiency and enable better exploration of the solution space despite the non-stationarity of the IoT environment. MMADDPG and MADDPG converge more slowly, with converged rewards of approximately 18000 and 15000, respectively. Both algorithms experience a significant drop in rewards around 2000 episodes.

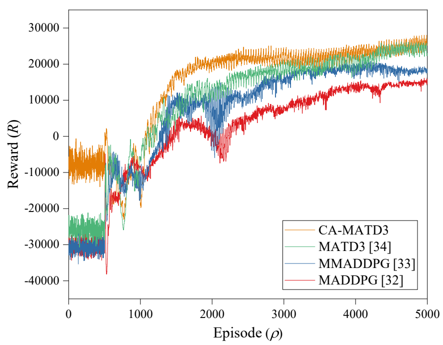


Fig. 10: Rewards of different algorithms.

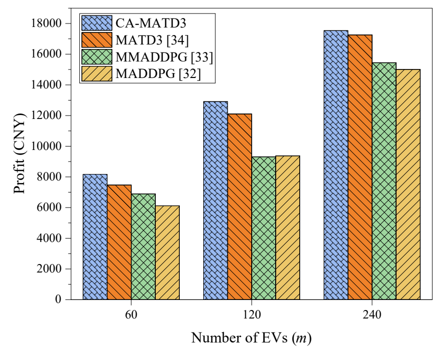


Fig. 11: Overall profit of the CSA with different algorithms under different EV scales.

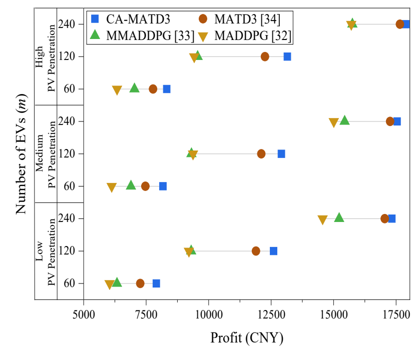


Fig. 12: Overall profit of the CSA with different algorithms in various scenarios.

TABLE III: Convergence performance of the algorithms with different EV scales (episode)

EV scale	CA-MATD3	MATD3 [34]	MMADDPG [33]	MADDPG [32]
small (m = 60)	2200	2860	3510	3670
medium (m = 120)	2430	3220	3240	3910
large (m = 240)	2690	3420	4050	4270

This is because these algorithms employ only one critic network, leading to an accumulation of Q-value estimation bias during training, which causes an overestimation of the values of certain actions and affects the stability of the training process. Compared with other benchmark algorithms (i.e., MADDPG, MMADDPG and MATD3), CA-MATD3 can achieve faster convergence while improving the convergence reward by 66.7%, 38.9%, and 4.2%, respectively.

To test the adaptability of CA-MATD3 in handling uncertain scenarios, we evaluate the convergence performance of different algorithms under various EV scales (i.e., small, medium and large) are established, where the number of EVs (i.e., m) is 60, 120 and 240, respectively.

Table III exhibits the convergence performance of the algorithms for finding a better solution with different EV scales. For the small-scale scenario, CA-MATD3 converges after about 2200 episodes, which is 40.1%, 37.3%, and 23.1% faster than MADDPG, MMADDPG, and MATD3, respectively. CA-MATD3 has the best performance, mainly due to the introduction of the cosine annealing method, which significantly improves the algorithm's efficiency. As the EV scale increases from small to medium and large, the convergence time of MATD3 increases by 360 and 560 episodes, respectively. In contrast, the convergence time of CA-MATD3 only increases by 230 and 470 episodes, respectively, which does not increase abruptly with the number of EVs. For the large-scale scenario, CA-MATD3 still achieves the fastest convergence, which is 37.0%, 33.6%, and 21.3% faster than MADDPG, MMADDPG, and MATD3, respectively. This indicates that CA-MATD3 has a stronger adaptability to the expansion of the EV scale.

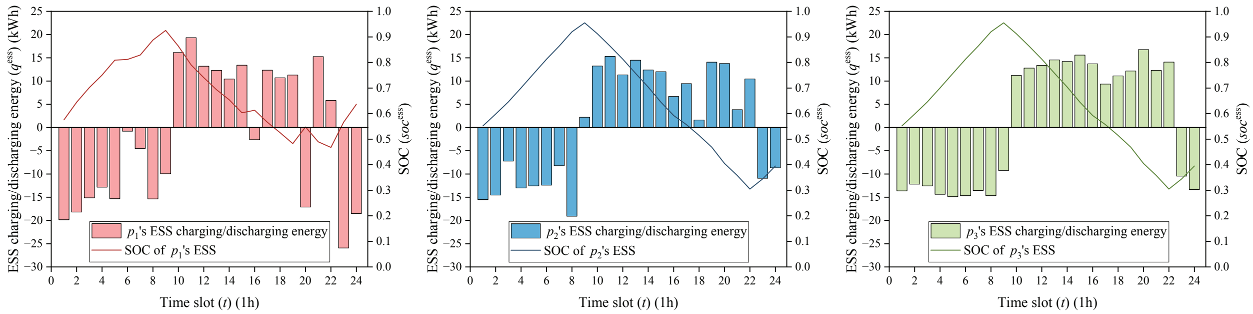
E. Effectiveness Comparison

In order to evaluate the effectiveness of CA-MATD3 in maximizing the overall profit of the CSA under different scenarios, we also conduct experiments with different EV scales (i.e., small, medium and large).

Fig. 11 displays the overall profit of the CSA with different algorithms under different EV scales (i.e., small, medium and large). From the overview in Fig. 11, it is obvious that the overall profit of the CSA shows a monotonically increasing trend as the network scale increases and CA-MATD3 has the best performance in all scales.

For the small-scale scenario, CA-MATD3 improves the overall profit of the CSA by 33.5%, 18.6%, and 9.4%, respectively, compared with MADDPG, MMADDPG, and MATD3. This is mainly due to its improvement in exploring the solution space, resulting in finding a better strategy for the CSA. For the medium-scale scenario, CA-MATD3 outperforms MADDPG, MMADDPG, and MATD3 by 37.7%, 38.6%, and 6.6%, respectively. The profit gap between MADDPG, MMADDPG, and CA-MATD3 widens as the scale increases, which indicates that CA-MATD3 shows better adaptation to the expansion of the solution space. As the EV scale increases, CA-MATD3 enables a better solution that significantly enhances the overall profit of the CSA as the energy demand of EVs increases. For the large-scale scenario, CA-MATD3 achieves a profit increase of 16.9%, 13.6%, and 1.6%, respectively, compared with MADDPG, MMADDPG, and MATD3. Although CA-MATD3 still achieves the best performance in the large-scale scenario, the profit gap between CA-MATD3 and the other benchmark algorithms has reduced. This is mainly due to the limited number of charging piles and the physical energy supply capacity of edge nodes. As a result, PSCSs cannot accommodate the additional energy demand of EVs, which limits further increases in the overall profit of the CSA.

To further assess CA-MATD3's robustness against the physical uncertainty in renewable energy generation, we extend the experimental scope to include scenarios with different



(a) PSCS1 (p_1) (b) PSCS2 (p_2) (c) PSCS3 (p_3)

Fig. 13: PSCS's ESS charging/discharging actuation with CA-MATD3.

PV penetration levels. The PV generation shown in Fig. 4 represents the medium PV penetration scenario. Relative to this baseline, the high PV penetration scenario assumes a 50% increase in PV output, while the low PV penetration scenario assumes a 50% decrease.

Fig. 12 presents the overall profit of the CSA with different algorithms in various scenarios, where CA-MATD3 consistently achieves the best performance. In the low PV penetration scenario, CA-MATD3 improves the overall profit of the CSA by 8.9%, 5.9%, and 1.7% for small, medium, and large scales, respectively, compared to the second-best algorithm (i.e., MATD3). In the high PV penetration scenario, CA-MATD3 outperforms the second-best algorithm (i.e., MATD3) by 7.0%, 7.3%, and 1.3% for small, medium, and large scales, respectively. This consistent superiority is primarily attributed to the introduction of the cosine annealing method, which facilitates efficient exploration of the solution space across diverse scenarios, thereby enhancing CA-MATD3's robustness against fluctuations in distributed renewable generation and improving its adaptability to different PV penetration levels.

F. Energy Scheduling

Fig. 13 displays each PSCS's ESS charging/discharging energy actuation with CA-MATD3. Each PSCS's ESS remains in a charging state from 00:00 to 07:00, and is predominantly in a discharging state from 08:00 to 21:00. This is mainly because CA-MATD3 utilizes the feature of the TOU price of the grid by purchasing energy at low price and selling it at high price, thereby improving the overall profit of the CSA. Besides, by charging during off-peak periods (i.e., 00:00-07:00), the ESS is fully prepared to meet the energy demand of EVs during peak periods (i.e., 9:00-11:00, and 19:00-21:00). CA-MATD3 ensures that the ESS stores appropriate energy to flexibly handle EVs' energy demand.

Meanwhile, the SOC of each PSCS's ESS reaches a peak of approximately 0.95 during the charging process, and remains above the safe threshold of 0.2 during discharging. This indicates that CA-MATD3 can generate safe actuation commands to prevent both overcharging and overdischarging of the ESS, thereby extending the lifespan of the ESS batteries. At the end of the day, the ESS's SOC is stabilized at approximately 0.5. This ensures that the ESS has sufficient energy to satisfy unexpected energy demand of EVs, while also having enough capacity to store energy purchased when the grid's TOU price is low next day, thereby reducing the energy purchase cost.

V. Conclusions and Future Work

This paper has proposed a cyber-physical cooperative scheme based on the Cloud-Edge collaborative CA-MATD3 algorithm to maximize the overall profit of CSA through energy scheduling and dynamic pricing. In addition, the IDF is introduced to model the cyber-physical coupling between pricing signals and EV demand, effectively avoiding irrational monopoly pricing strategies among fully cooperative edge agents. Furthermore, the universal gravitation model is employed to formulate the spatial-temporal attraction of each PSCS to EVs. Extensive experimental results show that CA-MATD3 outperforms other benchmark algorithms in terms of both the overall profit of CSA and convergence efficiency under various EV scalability scenarios. The introduced IDF can effectively regulate the charging price of PSCSs. In particular, CA-MATD3 is able to effectively improve convergence efficiency in large-scale scenarios. Additionally, by decreasing urban PSCSs' charging price and slightly increasing suburban PSCSs' charging price, CA-MATD3 with IDF significantly enhances the profitability of the CSA.

In our future work, we will focus on the cross-domain transactive energy trading among multiple CSAs and the joint optimization between the power grid and edge nodes. Simultaneously, we will further investigate a cyber-physical joint optimization framework that incorporates dynamic user equilibrium within the transportation network, thereby more accurately capturing the coupled effects of traffic congestion on energy scheduling.

References

[1] S. K. Pata, U. K. Pata, and Q. Wang, "Ecological power of energy storage, clean fuel innovation, and energy-related research and development technologies," *Renewable Energy*, vol. 241, p. 122377, 2025.

[2] T. D. de Lima, J. Soares, F. Lezama, J. F. Franco, and Z. Vale, "A risk-based planning approach for sustainable distribution systems considering ev charging stations and carbon taxes," *IEEE Transactions on Sustainable Energy*, vol. 14, no. 4, pp. 2294–2307, 2023.

[3] G. Yanagawa and H. Aki, "Grid flexibility provision by optimization of fast-charging demand of battery electric vehicles," *IEEE Transactions on Smart Grid*, vol. 14, no. 3, pp. 2202–2213, 2022.

[4] I. E. Agency, "Global ev outlook 2024," International Energy Agency, Paris, France, Tech. Rep., 2024.

[5] C. Kong, B. P. Rimal, M. Reisslein, M. Maier, I. S. Bayram, and M. Devetsikiotis, "Cloud-based charging management of heterogeneous electric vehicles in a network of charging stations: Price incentive versus capacity expansion," *IEEE Transactions on Services Computing*, vol. 15, no. 3, pp. 1693–1706, 2020.
[6] B. Williams, D. Bishop, G. Hooper, and J. Chase, "Driving change: Electric vehicle charging behavior and peak loading," *Renewable and Sustainable Energy Reviews*, vol. 189, p. 113953, 2024.
[7] P. V. Dahiwale, Z. H. Rather, and I. Mitra, "A comprehensive review of smart charging strategies for electric vehicles and way forward," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 9, pp. 10 462–10 482, 2024.
[8] C. Ge, S. Lin, F. Li, P. Wang, F. Yang, and D. Li, "Optimal coordination method for an adn with multiple network-constrained vpps," *IEEE Transactions on Power Systems*, vol. 40, no. 1, pp. 394–407, 2025.
[9] Y. Li, W. Chang, and Q. Yang, "Deep reinforcement learning based hierarchical energy management for virtual power plant with aggregated multiple heterogeneous microgrids," *Applied Energy*, vol. 382, p. 125333, 2025.
[10] X. Kong, W. Lu, J. Wu, C. Wang, X. Zhao, W. Hu, and Y. Shen, "Real-time pricing method for vpp demand response based on per-ddpg algorithm," *Energy*, vol. 271, p. 127036, 2023.
[11] C. Luo, Y. Huang, and V. Gupta, "Dynamic pricing and energy management strategy for ev charging stations under uncertainties," in *Proceedings of the International Conference on Vehicle Technology and Intelligent Transport Systems - VEHITS*, INSTICC. SciTePress, 2016, pp. 49–59.
[12] 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, 2021.
[13] R. Zhang, H. Kan, and I. Moon, "Trip pricing in user-based relocation for station-based carsharing systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 1, pp. 591–603, 2025.
[14] N. Chen, M. Li, M. Wang, Z. Su, J. Li, and X. S. Shen, "A dynamic pricing based scheduling scheme for electric vehicles as mobile energy storages," in *ICC 2021-IEEE International Conference on Communications*. IEEE, 2021, pp. 1–6.
[15] W. Meng, D. Song, L. Huang, X. Chen, J. Yang, M. Dong, M. Talaat, and M. Elkholy, "Distributed energy management of electric vehicle charging stations based on hierarchical pricing mechanism and aggregate feasible regions," *Energy*, vol. 291, p. 130332, 2024.
[16] N. Mignoni, R. Carli, and M. Dotoli, "Distributed noncooperative mpc for energy scheduling of charging and trading electric vehicles in energy communities," *IEEE Transactions on Control Systems Technology*, vol. 31, no. 5, pp. 2159–2172, 2023.
[17] C. Park, H. Lee, W. J. Yun, S. Park, S. Jung, and J. Kim, "Poster: Coordinated multi-agent reinforcement learning for unmanned aerial vehicle swarms in autonomous mobile access applications," in *2023 IEEE 43rd International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2023, pp. 1043–1044.
[18] T. Qian, C. Shao, X. Li, X. Wang, Z. Chen, and M. Shahidehpour, "Multi-agent deep reinforcement learning method for ev charging station game," *IEEE Transactions on Power Systems*, vol. 37, no. 3, pp. 1682–1694, 2021.
[19] S. Lee and D.-H. Choi, "Three-stage deep reinforcement learning for privacy-and safety-aware smart electric vehicle charging station scheduling and volt/var control," *IEEE Internet of Things Journal*, vol. 11, no. 5, pp. 8578–8589, 2023.
[20] M. Chen, L. Yu, Z. Chen, T. Zhang, D. Qiu, Y. Ye, and M. Zhang, "Joint differentiated pricing and energy-carbon trading for electric vehicle charging stations: An admm-based nash bargaining solution," *IEEE Internet of Things Journal*, vol. 12, no. 24, pp. 55 693–55 707, 2025.
[21] Q. Zhang, T. Qin, J. Wu, R. Hao, X. Su, and C. Li, "Synergistic operation strategy of electric-hydrogen charging station alliance based on differentiated characteristics," *Energy*, vol. 304, p. 132132, 2024.
[22] M.-L. Arlt and N. Astier, "Do retail businesses have efficient incentives to invest in public charging stations for electric vehicles?" *Energy Economics*, vol. 124, p. 106777, 2023.
[23] H. Yu, H. Wang, D. Tiwari, J. Li, and S.-J. Park, "Stellaris: Staleness-aware distributed reinforcement learning with serverless computing," in *SC24: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 2024, pp. 1–17.
[24] S. Wu, Q. Wang, Q. Chen, C. Yu, and Y. Tang, "Cyber-physical integrated planning of distribution networks considering spatial-temporal flexible resources," *Protection and Control of Modern Power Systems*, vol. 9, no. 3, pp. 142–156, 2024.
[25] W. Wu, J. Zhu, Y. Liu, T. Luo, Z. Chen, and H. Dong, "A coordinated model for multiple electric vehicle aggregators to grid considering imbalanced liability trading," *IEEE Transactions on Smart Grid*, vol. 15, no. 2, pp. 1876–1890, 2023.
[26] J. Jung, S. Moon, S. Yeo, and D.-J. Lee, "How would the carbon market affect the choice of input factors for production? a duopolistic model," *Energy*, vol. 282, no. 1, p. 129024, 2023.
[27] L. Zhang, R. Shi, X. Ma, L. Jia, and K. Y. Lee, "Highway self-contained multi-microgrid energy management strategy based on universal gravitation," *Energy*, p. 136430, 2025.
[28] B. Chatterjee, V. Kungurtsev, and D. Alistarh, "Federated sgd with local asynchrony," in *2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2024, pp. 857–868.
[29] X. Lu, H. Zheng, W. Liu, Y. Jiang, and H. Wu, "Pop-fl: towards efficient federated learning on edge using parallel over-parameterization," *IEEE Transactions on Services Computing*, vol. 17, no. 2, pp. 617–630, 2024.
[30] C. S. Lai, D. Chen, J. Zhang, X. Zhang, X. Xu, G. A. Taylor, and L. L. Lai, "Profit maximization for large-scale energy storage systems to enable fast ev charging infrastructure in distribution networks," *Energy*, vol. 259, p. 124852, 2022.
[31] B. Dai, F. Wang, Y. Chang, F. Chu, and S. Song, "Process model-assisted optimization control method based on model predictive control and deep reinforcement learning for coordinated control system in coal-fired power unit," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 228–239, 2024.
[32] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 6382–6393.
[33] Y. Zhang, Q. Yang, D. An, D. Li, and Z. Wu, "Multistep multiagent reinforcement learning for optimal energy schedule strategy of charging stations in smart grid," *IEEE Transactions on Cybernetics*, vol. 53, no. 7, pp. 4292–4305, 2022.
[34] J. Ackermann, V. Gabler, T. Osa, and M. Sugiyama, "Reducing overestimation bias in multi-agent domains using double centralized critics," *arXiv preprint arXiv:1910.01465*, 2019.
[35] N. Aguiar, A. Dubey, and V. Gupta, "Pricing demand-side flexibility with noisy consumers: Mean-variance trade-offs," *IEEE Transactions on Power Systems*, vol. 38, no. 2, pp. 1151–1161, 2022.
[36] W. Guo, Z. Chang, X. Guo, P. Wu, and Z. Han, "Incentive mechanism for edge computing-based blockchain: A sequential game approach," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 11, pp. 7899–7909, 2022.
[37] L. Pavel, "Distributed gne seeking under partial-decision information over networks via a doubly-augmented operator splitting approach," *IEEE Transactions on Automatic Control*, vol. 65, no. 4, pp. 1584–1597, 2020.