Personalized preference based electric vehicle charging recommendation considering photovoltaic consumption: A transfer reinforcement learning method

Wenlei Chen, Di Liu, Junwei Cao, *Senior Member, IEEE*

***Abstract*—The electrification of transportation has emerged as a notable trend due to advancements in battery technology and the widespread adoption of renewable energy such as photovoltaics (PVs). Many countries have instituted policies aimed at expediting the penetration of electric vehicles (EVs). However, prolonged charging queues and the failure to adequately meet personalized preferences for charging price and time cost have significantly impacted the user experience, thereby impeding the broader adoption of EVs. Moreover, the potential of EVs to use the power from PV panels at charging stations for lower charging tariffs through charging recommendation has not been fully explored. In this paper, we present a charging recommendation method to optimize the drivers' charging experience, offering three recommendation modes: time priority, price priority and balanced to enhance the compliance level of recommendations. We also consider PV generation in the recommendations, enabling drivers to obtain lower charging tariffs while promoting PV consumption. We formulate the problem as a Markov Decision Process (MDP) and design a customized reinforcement learning (RL) method. Extensive simulations are conducted using the SUMO simulation platform. Results indicate that compared to existing methods, our method promotes the PV consumption ratio by 10.5% and effectively enhances the Quality of Experience (QoE), thereby increasing the recommendation compliance level by 17.4%.**

***Index Terms*—Charging recommendation, electric vehicles, PV consumption, personalized preference, transfer reinforcement learning.**

NOMENCLATURE

$\mathcal{A}$	Action space.
$\mathcal{D}$	Set of EVs that send charging requests.
$\mathcal{G}$	Weighted directed graph that represents the road network.
$\mathcal{H}$	Set of charging stations.
$\mathcal{P}$	Transition function.
$\mathcal{R}$	Reward function.
$\mathcal{S}$	State space.
$\boldsymbol{C}$	Charging price at all stations.
$\boldsymbol{N}_t^{\mathrm{EV}}$	Number of EVs that are queuing at charging stations and are about to arrive at charging stations at time instant t.
$\boldsymbol{N}^{\mathrm{p}}$	Number of charging poles in charging stations.
$\boldsymbol{T}_t^{\mathrm{r}}$	Remaining charging time of charging poles at time instant t.
$\boldsymbol{T}_i^{\mathrm{tr}}$	Travel time of the EV i to all stations.
$\boldsymbol{T}_i^{\mathrm{W}}$	Waiting time of the EV i at all stations.
$\boldsymbol{T}_i^{\mathrm{total}}$	Time from when the EV i sends a charging request to when it receives charging service at all stations.
a_t	Action taken at the environment step t.
c_j	Charging price at the station j.
e_i	Battery capacity of the EV i.
$f_{j,t}$	Binary variable which indicates whether there is surplus PV power at the station j at time instant t.
$l_{j,t}$	Charging load at the station j at time instant t.
$p_{j,t}^{\mathrm{gen}}$	Power generated by PV panels at the station j at time instant t.
$p_{j,t}^{\mathrm{surplus}}$	Surplus power of the PV panels at the station j at time instant t.
p_i^{EV}	Average power consumption of the EV i during driving.
r_{t+1}	Instantaneous reward.
s_t	State at the environment step t.
t^{ch}	Charging duration of EVs.
$t_{i,j}^{\mathrm{total}}$	Time spent from when the EV i sends a charging request to when it receives the charging service at the station j.
$t_{i,j}^{\mathrm{tr}}$	Travel time of the EV i to the station j.
$t_{i,j}^{\mathrm{w}}$	Waiting time of EV i at the station j.
w_i^{t}	Weight coefficient of time.
w_i^{c}	Weight coefficient of price.
$x_{i,j}$	Binary decision variable which indicates whether the EV i is recommended to the station j.
$A(s_t, a_t)$	Advantage function.
G_t^{λ}	λ-return.
$Q(s_t, a_t)$	Action value function.
$QoE_{i,j}$	Quality of Experience of the EV i if charges at the station j.
$SoC_{i,t}$	State of charge of the EV i at time instant t.
SoC_i^{th}	Charging threshold of the EV i.
$V(s_t)$	State value function.

(Corresponding author: Junwei Cao.)

Wenlei Chen is with the Department of Automation, Tsinghua University, Beijing 100084, China.

Di Liu is with the Department of Electrical Engineering and the National Key Laboratory of New Power System Operation and Control, Tsinghua University, Beijing 100084, China.

Junwei Cao is with the Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China (e-mail:jcao@tsinghua.edu.cn).

B	Number of samples contained in a block.
M	Capacity of experience replay buffer.
S	Size of cache.
γ	Discount factor.
δ_t	TD error.
λ	Decay rate.
ω^{p}	Weight coefficient of PV consumption.

I. Introduction

A. Backgrounds & Motivations

In recent years, renewable energy has been playing an increasingly important role in the global energy structure. The electrification of transportation can help mitigate the differences between the supply and demand of renewable energy, providing a more environmentally friendly travel option. Therefore, it has been promoted in many countries and regions, becoming a significant trend. The report shows that total sales of electric vehicles (EVs) in China reached 6.9 million in 2022, a year-on-year increase of 97.1% [1].

However, some consumers still opt for traditional internal combustion vehicles due to the extra time spent on charging. Charging scheduling and charging recommendation are methods to address this issue. Previous research on charging scheduling mainly focused on when and at how much power EVs in a charging station (CS) should be charged [2], [3]. In contrast, charging recommendation focuses on vehicles in transit, helping drivers in determining the most suitable CS for charging.

Due to the mismatch between charging demand and the spatial distribution of CSs, CSs in hotspots often experience queues. In such cases, recommending EV drivers to go to slightly farther CSs where no queues are present can reduce the time cost, thus improving the user experience. However, both past and future charging requests from other vehicles can affect the current vehicle's queueing time at the CS, making it challenging to estimate queueing time.

Quality of Experience (QoE) describes how users perceive the quality of an application [4]. For charging recommendation services, the charging price and the time cost before accessing charging poles affect the drivers' QoE. Different EV drivers exhibit distinct preferences for time cost and charging price. Therefore, the charging recommendation system should consider the needs of different groups to elevate the drivers' QoE. However, existing research mainly concentrates on optimizing time-related metrics, without fully considering the QoE.

With the price reduction of photovoltaic (PV) panels, many CSs are equipped with PV panels. However, the volatility and intermittency of PVs make PV consumption challenging. To this end, CSs can offer discounts on charging prices to attract EV drivers when there is surplus PV power available. Existing research on charging scheduling addresses the PV consumption issue within individual CSs. However, in the field of charging recommendation, the mobility of EVs isn't fully leveraged to address the overall PV consumption problem across all CSs within a region. This may lead to suboptimal PV consumption and the charging economy from a global perspective.

Charging recommendation is a sequential decision-making problem, which faces a dynamic environment and is difficult to be described by an explicit model. Recently, reinforcement learning (RL) has shown great potential in solving such problems, so it is intuitive to design a charging recommendation method based on RL. However, there are some challenges: First, the results of the charging recommendation can only be known after EVs receive charging services, so the reward is delayed. In scenarios with few vehicles requesting charging, the reward is sparse. Classical RL algorithms such as Deep Q network (DQN) suffer from low learning efficiency, which is more pronounced under delayed and sparse rewards. Second, when the scenario changes, the performance of models trained in limited scenarios will degrade, and the time cost of retraining the model from scratch is significant.

B. Contributions

To address the aforementioned issues, this paper proposes a transfer reinforcement learning based charging recommendation method that takes into account the diversity of drivers' preferences, offers drivers three options and considers the PV consumption at CSs. The contributions of this paper are outlined as follows:

- We consider the diverse preferences of EV drivers for time cost and charging price, and provide three modes: time priority, price priority, and balanced, to improve the drivers' QoE. Additionally, PV power consumption is considered in the charging recommendation problem.
- In order to improve the learning efficiency of the agent, in the reinforcement learning algorithm, we introduce the prioritized cache construction mechanism and λ-return.
- A transferable charging recommendation framework is designed by integrating our customized RL algorithm with transfer learning to accelerate learning in different scenarios.

II. Related Work

In the context of the increasing penetration of EVs, the mismatch between charging demand and the supply of charging services is becoming increasingly prominent.

Many scholars have explored charging scheduling methods to address this issue. The work in [5] provides a detailed review of advancements in the field of EV scheduling from the perspectives of single- and multiple-CSs, and EV aggregator levels. The work in [6] proposes a customized actor-critic method to reduce the charging cost of EV fleets while shaving peak load. The work in [7] also put forward a model-free RL based method to optimize charging cost through bidirectional power flow. The literature mentioned above assumes that charging power is continuously variable, while the work in [8] introduces an on-off charging strategy, where the charging poles can only switch between two states. The work in [9] provides a different view, it analyzes the factors that affect the development of CSs and matches charging demand with supply by influencing the accessibility of CSs.

Different from the above charging scheduling methods which focus on parked EVs, charging recommendation provide

CS suggestions for on-the-move EVs, serving as a complementary solution to this issue. The work in [10] designs a minimum waiting time CS recommendation policy based on vehicular networks, but this method can't provide an accurate estimate of charging waiting time due to the lack of information on the number of EVs heading to CSs. On this basis, a reservation mechanism is introduced in [11] to improve the accuracy of estimated charging waiting time. Many advanced charging recommendation strategies have been further explored. A deadline-driven charging recommendation algorithm is developed in [12] to minimize waiting time and increase the number of fully charged EVs. In [13], a Lyapunov optimization method based charging recommendation strategy is proposed to reduce the time from requesting the charging service to accessing it. The work in [14] develops a real-time charging recommendation system for electric taxis via data mining, which can assign CSs with minimal time cost to taxis. In [15], Q-learning is utilized to develop a charging recommendation method that optimizes both the cost of traveling to CSs and the cost of charging. Although these user-oriented methods optimize either time-related or cost-related metrics, the diversity in charging needs among different groups is overlooked.

Some studies have noticed the negative experiences that arise from only considering one factor when recommending CSs to EV drivers. In [16]–[18], game theory, multi-agent reinforcement learning (MARL) and heuristic method are adopted to develop multi-objective charging recommendation strategies respectively. They all consider objectives related to time and cost and balance the two objectives using fixed weights. The advantage lies in its simplicity. However, it does not take the diversity of user preferences into account. The work in [19] provides three metrics of time, cost, and distance, allowing EV drivers to adjust the weights of these metrics. Although this approach provides great flexibility to EV drivers and covers different preferences, it may not be practical to require drivers to set these weights manually.

Many CSs are equipped with PV panels. If utilized effectively, PVs can reduce charging costs, enhance QoE, and promote PV consumption. Existing research aims to solve the problem of PV consumption at individual CSs through charging scheduling [20]–[23]. In [21], a combination of PV output prediction methods and state of charge (SoC) based charging strategies is proposed to enhance PV energy harvest. Some research on charging recommendation has noticed the threat posed by the random distribution of charging demand in time and space to the stable operation of the distribution system [24], [25], but none of these studies have further explored the role of charging recommendation in reducing charging costs and promoting PV consumption. This gap in the literature serves as one of the motivations for our research.

Most of the aforementioned research adopts heuristic or optimization algorithms, but these algorithms are usually inefficient and cannot provide real-time responses to a large number of charging requests. Fast forward reasoning speed of RL makes it suitable for charging recommendation. In [26], DQN is adopted to minimize the total travel time. However, the classical DQN suffers from low learning efficiency, especially in cases of sparse rewards and delayed rewards [27], [28]. Some researchers have designed algorithms specifically for handling delayed rewards. In [29], the agent is trained by decomposing return to obtain immediate reward, but this algorithm is not stable enough to apply to the charging recommendation problem. In [30], the authors propose the Delay-Correcting Actor-Critic method to deal with random delays in the environment, but the assumption of the possible maximum delay of observations and actions is not obtainable in charging recommendation. In [31], the authors combined the λ-return with DQN. Since this algorithm incorporates a longer time horizon in the credit assignment process, it can help the agent learn more efficiently in scenarios with delayed or sparse rewards. Therefore, we adopt this algorithm.

Charging recommendation has to deal with dynamic and changing environments, so knowledge transfer is crucial. Previous research has explored diverse methods, including lifelong learning [32], knowledge distillation [33], and transfer learning [34]. Transfer learning can transfer knowledge at different levels, such as demonstration trajectories, model dynamics, policies, value functions, etc. [35]. Its flexibility enables agents to swiftly adapt to new environments, making it suitable for charging recommendation.

The surveyed articles that focus on charging recommendation are shown in Table I.

TABLE I
SUMMARY OF RELATED WORKS

Number	PV	Time cost	Charging price	Personalized preference	Method
[10]		✓			Heuristic method
[11]		✓			Heuristic method
[12]		✓			Heuristic method
[13]		✓			Lyapunov optimization method
[14]		✓			Data mining
[15]			✓		Reinforcement learning
[16]		✓	✓		Game theory
[17]		✓	✓		Reinforcement learning
[18]		✓	✓		Heuristic method
[19]		✓	✓	✓	Stochastic method
[24]		✓			Graph reinforcement learning
[25]		✓			Online learning method
[26]		✓			Reinforcement learning
Ours	✓	✓	✓	✓	Transfer reinforcement learning

III. PROBLEM FORMULATION

In this section, we will elaborate on the issues involved in the charging recommendation system.

The overall architecture of the charging recommendation system is depicted in Fig. 1. It consists of three main participants.

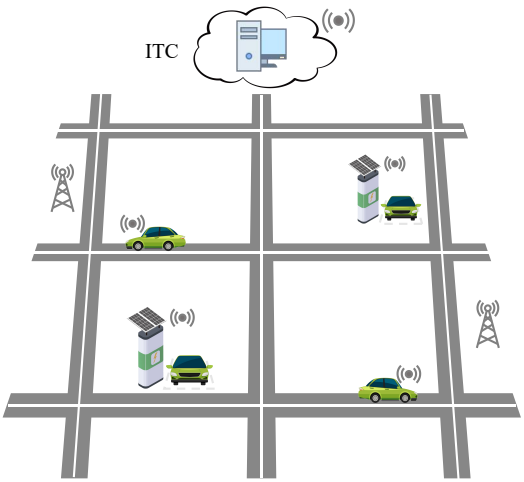


Fig. 1. Overall architecture of the charging recommendation system.

1) EV: Each EV is equipped with an e_i kWh battery, with a state of charge of $SoC_{i,t} \in [0,1]$. During the journey, if the SoC of an EV falls below the charging threshold SoC_i^{th}, EV drivers can choose one of three modes: time priority, price priority, or balanced, based on their preferences for charging time and price. Subsequently, a charging request is sent out. Define the set of EVs that send charging requests as $\mathcal{D}$, which has a cardinality of D. The average power consumption of an EV is denoted as p_i^{EV} kW.

2) CS: Define the set of CSs as $\mathcal{H}$, which has a cardinality of H. Each CS is equipped with $\boldsymbol{N}^{\text{p}}$ charging poles. Each CS records the remaining charging time $\boldsymbol{T}_t^{\text{r}}$ of each charging pole and maintains a list of EVs queueing for charging at the station as well as EVs that are about to arrive for charging, with a total number of $\boldsymbol{N}_t^{\text{EV}}$. All CSs are equipped with PV panels, which generate power denoted as $p_{j,t}^{\text{gen}}$. The charging load at station j is denoted as $l_{j,t}$. The CS prioritizes using electricity from the PV panels to charge EVs. When the PV generation is insufficient, the remaining electricity is supplied by the grid. When there is surplus PV power, the CS offers discounted charging prices to incentivize EVs to participate in PV consumption.

3) Intelligent Traffic Center (ITC): It is represented by an agent that is capable of processing charging requests in a real-time manner. The road network can be seen as a weighted directed graph $\mathcal{G} = (\mathcal{E}, \mathcal{L})$, where $\mathcal{E}$ represents the set of roads and $\mathcal{L}$ represents the set of intersections. The weight assigned to each road indicates the average travel speed along that road. This graph is known to the ITC.

Upon receiving a charging request, the center can recommend a CS for the EV based on the real-time state of the road network and CSs. Once the EV receives the recommendation result, it proceeds to the recommended CS to charge for t^{ch} before continuing to its destination. Note that "real-time" is relative to the time cost measured in minutes and the delay of ITC is typically in the order of seconds or even milliseconds. Therefore, this delay is not considered.

In our settings, drivers are concerned about the charging price and the time cost. Therefore, if the ith vehicle is recommended to go to the jth CS, the corresponding QoE can be defined as:

$$QoE_{i,j} = 1 - \omega_i^{\text{t}} t_{i,j}^{\text{total}} / t^{\text{ref}} - \omega_i^{\text{c}} c_j / c^{\text{ref}} \tag{1}$$

where $t_{i,j}^{\text{total}}$ is the time spent from when the EV sends a charging request to when it receives the charging service at CS j. $t_{i,j}^{\text{total}}$ consists of two parts: $t_{i,j}^{\text{tr}}$ (the travel time of ith EV to the recommended CS) and $t_{i,j}^{\text{w}}$ (the waiting time of ith EV at the recommended CS). c_j denotes the charging price at CS j. Both t^{ref} and c^{ref} are reference values, which can be set according to historical data. w_i^{t} and w_i^{c} represent the corresponding weight coefficients. In the time priority, price priority and balanced mode, $(\omega_i^{\text{t}}, \omega_i^{\text{c}})$ is set to (1,0), (0,1) and (0.5,0.5), respectively.

Note that battery degradation is associated with charging power and cycle counts [36], [37], and does not impact the time cost or charging price, and therefore, it does not affect the QoE. Hence, it is not considered.

The charging recommendation, aimed at optimizing the QoE while mitigating PV curtailment, can be formulated as the following optimization problem:

$$\begin{aligned} \max \quad & \frac{(1-\omega^{\text{p}})}{D} \sum_{i\in\mathcal{D}} \sum_{j\in\mathcal{H}} x_{i,j} QoE_{i,j} \\ & + \frac{\omega^{\text{p}}}{H} \sum_{j\in\mathcal{H}} (1 - \int_0^T p_{j,t}^{\text{surplus}} dt / \int_0^T p_{j,t}^{\text{gen}} dt) \qquad (2) \\ \text{s.t.} \quad & \sum_{j\in\mathcal{H}} x_{i,j} = 1, \quad \forall i \in \mathcal{D} \qquad (3) \\ & x_{i,j} \in \{0,1\}, \quad \forall i \in \mathcal{D}, \forall j \in \mathcal{H} \qquad (4) \end{aligned}$$

where $x_{i,j}$ is a binary decision variable. When it equals to 1, it indicates that the ith EV is recommended to the jth CS, otherwise, it is equal to 0. w^{p} is used to balance two objectives and is determined by the ITC operator. $p_{j,t}^{\text{surplus}}$ represents the surplus power of the PV panels at jth CS at time t, which is defined as follows:

$$p_{j,t}^{\text{surplus}} = \max\{p_{j,t}^{\text{gen}} - l_{j,t}, 0\} \tag{5}$$

The second term in the objective function is referred to as the PV consumption ratio, indicating the average proportion of PV power consumed by EVs at each CS.

IV. Proposed Methodology

This section introduces the proposed charging recommendation method. First, we formulate the problem as a Markov Decision Process. Then, we detail the proposed RL method with λ-return and priority cache construction. Finally, we explain how transfer learning transfers knowledge between different scenarios.

A. Formulation of Markov Decision Process

In this scenario, the ITC observes the current state of the road network, CSs and EVs, selects an action from the action space based on this observation, and then receives a reward based on the outcome of its action in the environment. The

goal is to learn the optimal policy that maximizes the total reward over time. The definitions of the action space, state space, and rewards are as follows:

State space $\mathcal{S}$: The state at environment step t is denoted as s_t. When there is no charging request at a certain moment, we treat it as a state. When there are N_t charging requests from EVs at actual time t, we treat each charging request as a state. Therefore, the subscript t in s_t refers to the environment step, not the actual time.

The states should be informative enough for the ITC to make decisions. They should include information from three aspects: EVs, road network, and CSs. It is very challenging for a neural network to directly extract useful information for decision-making from these raw inputs. Therefore, it is necessary to perform feature extraction on the raw state information to obtain a high-level state representation.

The travel time to each CS $\boldsymbol{T}_i^{\mathrm{tr}} = \{t_{i,1}^{\mathrm{tr}}, \cdots, t_{i,H}^{\mathrm{tr}}\}$ can be estimated using the Dijkstra algorithm as follows:

$$\boldsymbol{T}_i^{\mathrm{tr}} = \mathrm{Dijkstra}(\mathcal{G}) \tag{6}$$

Inspired by [24], we assume EVs that sent charging requests before the current one will arrive at the CSs ahead of the current one. Therefore the waiting time at each CS $\boldsymbol{T}_i^{\mathrm{w}} = \{t_{i,1}^{\mathrm{w}}, \cdots, t_{i,H}^{\mathrm{w}}\}$ can be estimated as follows:

$$\boldsymbol{T}_i^{\mathrm{W}} = \min_{\boldsymbol{N}_t^{\mathrm{EV}} \% \boldsymbol{N}^{\mathrm{P}} + 1} \boldsymbol{T}_t^{\mathrm{r}} + t^{\mathrm{ch}} \cdot \mathrm{floor}\left(\boldsymbol{N}_t^{\mathrm{EV}} / \boldsymbol{N}^{\mathrm{P}}\right) \tag{7}$$

Here, the floor function denotes rounding down to the nearest integer. Equation (7) represents selecting the $\boldsymbol{N}_t^{\mathrm{EV}} \% \boldsymbol{N}^{\mathrm{P}} + 1$th values after arranging $\boldsymbol{T}_t^{\mathrm{r}}$ in ascending order.

To facilitate the agent's perception of whether there is surplus power from the PV panels of each CS, a binary variable $f_{j,t}$ is defined as follows:

$$f_{j,t} = \begin{cases} 1, & p_{j,t}^{\mathrm{surplus}} \neq 0 \\ 0, & p_{j,t}^{\mathrm{surplus}} = 0 \end{cases} \tag{8}$$

The states can be defined as follows:

$$s_t = \begin{cases} \left[0, \boldsymbol{T}^{\mathrm{ref}}, 0, \boldsymbol{C}^{\mathrm{ref}}, 1, \boldsymbol{0}, \omega_i^{\mathrm{c}}, \omega^{\mathrm{P}}\right], & \text{no request} \\ \left[t^{\mathrm{ref}}, \boldsymbol{T}_i^{\mathrm{total}}, c^{\mathrm{ref}}, \boldsymbol{C}, 0, \boldsymbol{F}_{\boldsymbol{T}_i^{\mathrm{total}}}, \omega_i^{\mathrm{c}}, \omega^{\mathrm{P}}\right], & \text{otherwise} \end{cases} \tag{9}$$

where

$$\boldsymbol{T}_i^{\mathrm{total}} = \boldsymbol{T}_i^{\mathrm{w}} + \boldsymbol{T}_i^{\mathrm{tr}} \tag{10}$$

$$c^{\mathrm{ref}} = \max\{c_1, \cdots, c_H\} \tag{11}$$

$$\boldsymbol{C} = [c_1, \cdots, c_H] \tag{12}$$

$$\boldsymbol{F}_{\boldsymbol{T}_i^{\mathrm{total}}} = \left[f_{1,t_{i,1}^{\mathrm{total}}}, \cdots, f_{H,t_{i,H}^{\mathrm{total}}}\right] \tag{13}$$

Here, t^{ref} is a sufficiently large reference value to make the agent aware that the time cost of choosing the corresponding action is very high. $\boldsymbol{T}^{\mathrm{ref}}$ and $\boldsymbol{C}^{\mathrm{ref}}$ are the vector forms of t^{ref} and c^{ref}, respectively, each containing H identical elements. $\boldsymbol{F}_{\boldsymbol{T}_i^{\mathrm{total}}}$ represents whether there is surplus PV power at each CS at the estimated time of accessing the charging pole.

Note that in the above state representation, we have augmented the information of time, price, and surplus power. Each aspect of information contains $H+1$ elements instead of H, corresponding to the action of not recommending any CS and the H actions of recommending to H CSs. This design ensures that the agent has full knowledge of all actions.

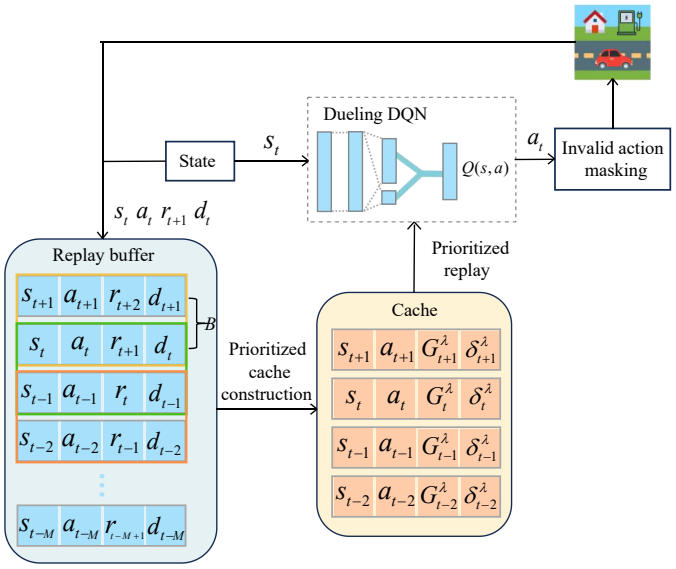


Fig. 2. Overall architecture of the proposed RL method.

Action space $\mathcal{A}$: The action space consists of one action of not recommending any CS and H actions of recommending to H CSs.

Reward function $\mathcal{R}$: The reward should reflect our optimization objective and therefore is defined as follows:

$$r_{t+1} = \begin{cases} 0, & \text{no request} \\ -(1-\omega^{\mathrm{P}}) r^{\mathrm{EV}} - \omega^{\mathrm{P}} r^{\mathrm{PV}}, & \text{otherwise} \end{cases} \tag{14}$$

$$r^{\mathrm{EV}} = \omega_i^{\mathrm{c}} \boldsymbol{C}[a_t] / c^{\mathrm{ref}} + \omega_i^{\mathrm{t}} \boldsymbol{T}_i^{\mathrm{total}}[a_t] / t^{\mathrm{ref}} \tag{15}$$

$$r^{\mathrm{PV}} = \begin{cases} 0, & \boldsymbol{F}_{\boldsymbol{T}_i^{\mathrm{total}}}[a_t] = 1 \\ 1, & \boldsymbol{F}_{\boldsymbol{T}_i^{\mathrm{total}}}[a_t] = 0 \end{cases} \tag{16}$$

Note that since the actual driving time and waiting time cannot be known in advance, we use their estimated values instead.

Transition function $\mathcal{P}$: In this scenario, the state of the environment is influenced not only by the previous state and the action taken but also by exogenous random factors such as randomly appearing new EVs and fluctuations in the power generated by the PV panels. We denote these random factors as ϵ_t. Therefore, we have:

$$s_{t+1} = f(s_t, a_t, \epsilon_t) \tag{17}$$

B. Reinforcement Learning Algorithm

Charging recommendation is a sequential decision-making problem, which is suitable to be addressed by RL. Monte Carlo (MC) and Q-learning are the most commonly used RL methods. However, MC has the drawback of greater variance in estimation, requiring more samples to converge, while Q-learning suffers from estimation bias. λ-return, as an

interpolation of the two, combines the advantages of both, defined as follows:

$$G_t^\lambda \doteq (1-\lambda)\sum_{n=1}^{T-t-1}\lambda^{n-1}G_{t:t+n} + \lambda^{T-t-1}G_t \tag{18}$$

$$G_{t:t+n} = r_t + \cdots + \gamma^{n-1}r_{t+n-1} + \gamma^n \max_{a'} Q(s_{t+n}, a') \tag{19}$$

where $\gamma \in [0,1]$ represents the discount factor, and $\lambda \in [0,1]$ is decay rate.

In [31], a recursive computation method is provided to reduce time complexity:

$$G_t^\lambda = G_{t:t+1} + \gamma\lambda[G_{t+1}^\lambda - \max_{a'} Q(s_{t+1}, a')] \tag{20}$$

The update formula for the estimated Q-function is:

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha\left[G_t^\lambda - Q(s_t, a_t)\right] \tag{21}$$

The state space is typically very large. To generalize over states, a neural network is often used to approximate the Q-function instead of a Q-table. When training the Q network, to improve sampling efficiency, we use an experience replay buffer to store samples. The capacity of the experience replay buffer can reach millions, but during each training iteration of the agent, only a subset of this data is used. It would be computationally expensive if we were to recursively compute the λ return for all samples in the entire experience replay buffer. To address this issue, following [31], we introduce a cache with a size S much smaller than that of the experience replay buffer. Before each training epoch, we extract $\frac{S}{B}$ blocks from the experience replay buffer and place them into the cache. Each block consists of B consecutive samples, and according to Equation (20), we calculate the corresponding λ returns for each sample. During the training of the agent, we only extract data from this cache.

An additional benefit of using a cache is that since the target is the λ-return, which is computed before each epoch and remains constant during the training process, it serves as the target network. Therefore, only a policy network is needed in the DQN(λ) algorithm.

In the charging recommendation problem, there are no EVs requesting charging most of the time, so the actions that do not recommend any CS take the majority of experience replay buffer. To facilitate the learning of actual recommendations, inspired by [24], when constructing the cache, data blocks containing more actual recommendation actions are sampled with a higher probability. Assuming the capacity of the experience replay buffer is M, blocks are constructed in a sliding window manner, with each sliding step being 1. There are a total of $M-B+1$ data blocks to be sampled in the experience replay buffer. The proportion of actual recommendation actions in each data block B_i can be calculated as follows:

$$k(B_i) = 1 - \frac{N_i^{\text{nr}}}{B} \tag{22}$$

where N_i^{nr} represents the number of actions in B_i that do not recommend any CS.

The probability of the data block B_i being selected is defined as:

$$P(B_i) = \begin{cases} \frac{1+p}{M-B+1} & \text{if } k(B_i) > k_{\text{median}} \\ \frac{1}{M-B+1} & \text{if } k(B_i) = k_{\text{median}} \\ \frac{1-p}{M-B+1} & \text{if } k(B_i) < k_{\text{median}} \end{cases} \tag{23}$$

where p is a hyperparameter used to adjust the probability and k_{median} denote the median of all $k(B_i)$ values.

To further enhance the performance, two improvements of DQN are also applied in our algorithm: Dueling Q Networks [38] and Prioritized Experience Replay (PER) [39].

In PER, experiences with larger TD errors are sampled with higher priority to ensure the agent focuses more on learning from its mistakes. TD error is defined as follows:

$$\delta_t = G_t^\lambda - Q(s_t, a_t) \tag{24}$$

The probability of sample w_i being selected is defined as:

$$P(w_i) = \begin{cases} \frac{1+p}{S} & \text{if } |\delta_i| > \text{median}\left(|\delta_1|, \cdots, |\delta_S|\right) \\ \frac{1}{S} & \text{if } |\delta_i| = \text{median}\left(|\delta_1|, \cdots, |\delta_S|\right) \\ \frac{1-p}{S} & \text{if } |\delta_i| < \text{median}\left(|\delta_1|, \cdots, |\delta_S|\right) \end{cases} \tag{25}$$

The Dueling Q network is separated into two streams: the value stream that estimates the value function of being in a particular state and the advantage stream that estimates the advantage of taking each action in that state. The Q value is calculated by combining the two streams using the following equation:

$$Q(s_t, a_t) = V(s_t) + A(s_t, a_t) - \frac{1}{|\mathcal{A}|}\sum_{a'} A(s_t, a') \tag{26}$$

The agent can make mistakes sometimes. To prevent the agent from making incorrect actions that could disrupt the system's normal operation, for example: recommending any CS when no EVs are requesting charging, or not recommending any CSs when EVs are requesting charging, we draw inspiration from [40] and design an invalid action masking mechanism, defined as follows:

$$a_t = \begin{cases} a_{nr}, & \text{no EV request} \\ \underset{a \neq a_{nr}}{\arg\max}\, Q(s_t, a), & \text{otherwise} \end{cases} \tag{27}$$

The overall architecture of our RL method is shown in Fig. 2. d_t is a flag used to indicate whether an episode has ended. In the diagram of the replay buffer, the samples enclosed in boxes of different colors represent different consecutive data blocks.

C. Transfer Learning

Transfer learning (TL) aims to leverage knowledge from the source domain to improve the learning performance in a target domain. The source domain refers to the scenario used for pre-training models. The model trained in the source scenario will be transferred to a new scenario, that is, the target scenario, for fine-tuning to obtain the model we need. In previous research, RL-based charging recommendation algorithms were trained and tested in the same scenario. If the scenario changes, such as a change in traffic flow, a new model would need to be trained from scratch, which consumes a lot of computing

power, especially when the model is large. As shown in Fig. 3, in our proposed method, the weights of the action value network in the target domain are initialized to the same values as those in the source domain. Then, the charging recommendation continues in the new scenario, with fine-tuning of the action-value network. The data in the replay buffer and cache are discarded to expedite the agent's adaptation to the new scenario.

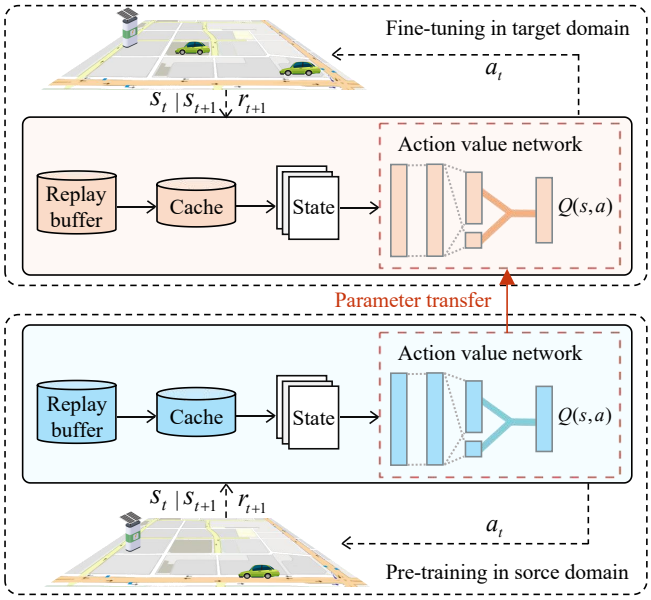


Fig. 3. Overview of the proposed transfer reinforcement learning method.

V. Simulation

A. Scenario and Parameter Settings

To simulate the driving and charging processes of EVs in the road network, SUMO [41] is used as the traffic simulation platform. The TRACI interface provided by SUMO allows us to obtain basic information about EVs, CSs, and the road network during the simulation process. It also enables the control of EVs to travel to designated CSs for charging. The simulation environment operates with a time step of 1 second. The charging recommendation method is implemented using Python-Pytorch and runs on a Ubuntu server equipped with an AMD Ryzen 7 3700X CPU and GeForce RTX 2080 GPU. Fig. 4 shows the simulation scenario, set in a region of the urban area of Beijing. In the road network, four CSs are set up. Each CS is set on the outermost lane of the road, located at the midpoint of the road, and is connected to the main road by a single-lane, one-way road. The maximum speed limit on each road in the network is 50 km/h.

Within the road network, all vehicles are configured as EVs with identical specifications: the battery capacity is 20 kWh and the average power consumption during driving, p_i^{EV}, is 10kW. The initial SoC of the battery and the charging threshold follow uniform distributions in the ranges of $[0.2, 0.4]$ and $[0.14, 0.17]$, respectively. EV trips are generated randomly using the script provided by SUMO. During the journey towards the destination, if the SoC falls below the threshold, a charging request is issued. Subsequently, the ITC recommends a suitable CS for the EV. The EV then follows the recommendation, proceeding to the recommended CS for a 10-minute charge before continuing toward its destination.

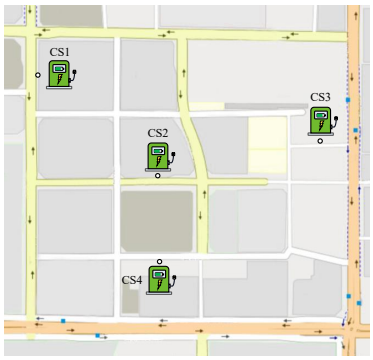


Fig. 4. The topology of the road network.

The charging prices at each CS are set at 1.3, 2, 1.4, and 1.7 yuan/kWh, respectively, and it is assumed that they remain constant throughout the entire simulation process. Since the area we investigate in the simulation is relatively small, for the sake of simplification, we reasonably assume that the PV panels at all charging stations generate the same power output. Here, the PV generation data we used is sourced from [42], collected during the summer of 2014 with measurements taken every minute. As the power output from PV panels fluctuates slowly, we assume that the power remains constant between adjacent data points.

The Dueling Q Network applied in our method consists of 2 fully connected layers, 1 value layer and 1 advantage layer. The input and output dimensions of each layer are illustrated in Table II. The relevant parameters of our method are listed in Table III. Each episode in the RL method corresponds to a simulation duration of 1h. The network is trained every 1800 environment steps.

TABLE II
The Structure of Dueling Q Nework

Layer name	Input dimensions	Output dimensions
1st fully connected layer	17	128
2nd fully connected layer	128	128
Value layer	128	5
Advantage layer	128	1

TABLE III
Parameters Related to The Method

Parameters	Descriptions	Values
lr	learning rate	10^{-4}
batch size	/	32
γ	discount factor	0.999
λ	decay rate	0.5
M	buffer size	10^5
S	cache size	1600
B	block size	200
p	priority factor	0.1

B. Simulation Results

In this subsection, we evaluate the performance of our method.

1) Validation of proposed RL algorithm: We design a scenario where an EV is inserted into the road network every 5 seconds (vehicle insertion rate as a shorthand), with the three modes of recommendation randomly selected. We use PV generation data from August 13th, 13:00-14:00, with ω^{P} set to 0.5. Under this scenario, we train the agent using 5 random seeds. The average return curves are shown in Fig. 5. Here, Prioritized Cache Construction is abbreviated as PCC. DQN refers to the version with Dueling Q Network and Prioritized Experience Replay. DQN+λ-return+PCC refers to the proposed algorithm described in the previous sections. From Fig. 5, it can be seen that at the end of the training, our algorithm has the highest average return, followed by DQN+λ-return and DQN, indicating that λ-return and PCC both improve the algorithm's performance. Furthermore, our algorithm exhibits a faster learning rate compared to the version without PCC. This is because PCC can more effectively utilize the experiences from actual recommendations.

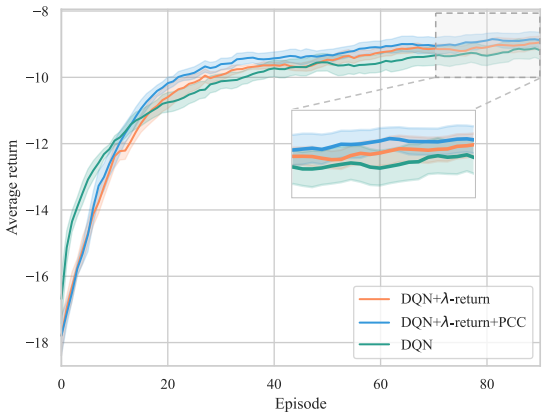


Fig. 5. Average return curves of three RL algorithms.

2) Effectiveness validation of transfer learning: In this part, we design a scenario where the vehicle insertion rate is 4. We use PV generation data from August 15th, 14:00-15:00, with ω^{P} set to 0.2. Due to the differences between this scenario and the one used in V-B1, the policy obtained from V-B1 is not entirely suitable for this scenario. Therefore, it is necessary to retrain the agent. The most straightforward method is to train a new agent from scratch in this scenario, as we do in V-B1. Another approach draws inspiration from transfer learning, where the neural network parameters obtained from V-B1 are transferred to the neural network being trained in the current scenario, followed by fine-tuning the network in this scenario. Fig. 6 illustrates the average return curves during the training process for the two training methods.

It can be observed that the initial policy with TL is near optimal, converges faster than learning from scratch, and outperforms in terms of the final average return. Therefore, TL can accelerate training and improve performance. Fig. 6 also illustrates that if the energy demand changes, models trained in other scenarios still perform well, indicating that the impact of energy demand changes on the method is limited.

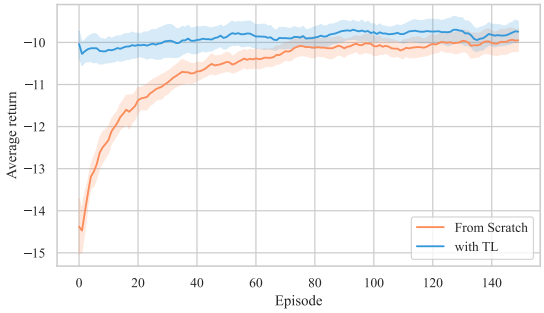


Fig. 6. The average return curves for the two training methods.

3) Impact on queue length: In this part, to validate the effectiveness of our method, we compare our method with the scheme of distance greedy method(i.e., selecting the nearest CS).

We use the model obtained from V-B1 as the initial model and fine-tune the model in a high-load scenario, where the vehicle insertion rate is 0.84, and the probabilities for the three optional modes are equal, with $\omega^{\mathrm{P}} = 0$.

We use the fine-tuned model to test our method in another high-load scenario (vehicle insertion rate of 0.84, with time priority, price priority, and balanced in the proportion of (0.9, 0.05, 0.05)) to evaluate its performance in improving the standard deviation of queued vehicles at each CS.

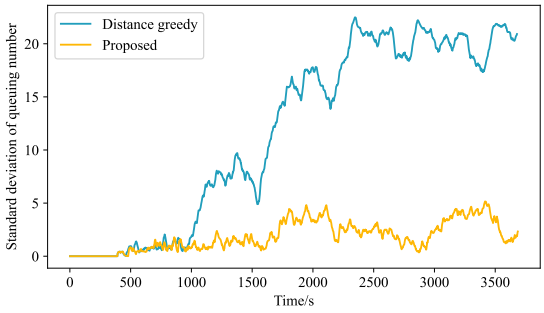
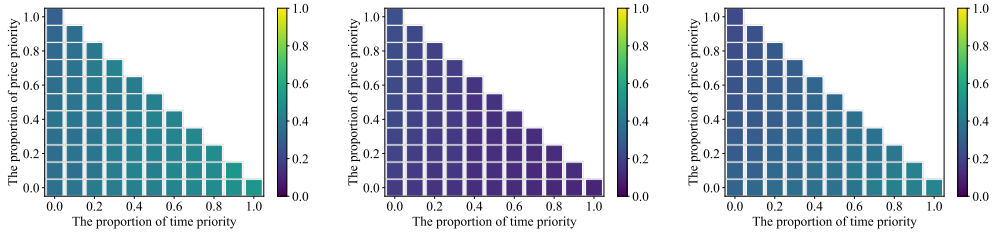


Fig. 7. Standard deviation of the number of vehicles queuing at each CS.

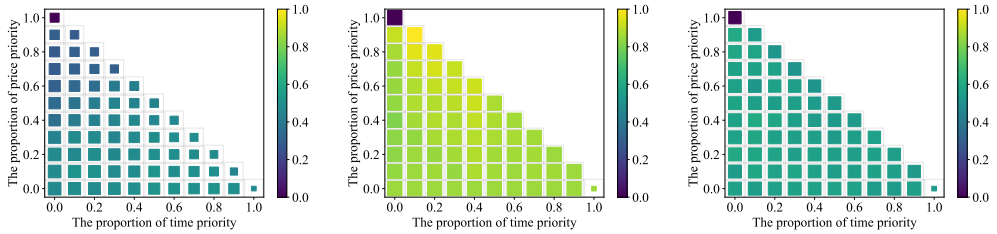
From the results in Fig. 7, it can be found that in high-load scenarios, the distance greedy method can lead to long queues at hot-spot stations, while others have fewer vehicles in line, resulting in a high standard deviation. This not only degrades the QoE of EV drivers but also contributes to traffic congestion. However, if the majority of drivers prefer optimizing their time experience, our method can quickly suppress the increase in standard deviation when it occurs, maintaining it at a lower level.

4) Impact on QoE: To further evaluate the effectiveness of our method in improving the QoE of EV drivers, we use the fine-tuned model to test the QoE under different combinations of recommendation mode proportions. The vehicle insertion rate remains at 0.84. We compare our method with the distance greedy method and the stochastic distributed method described



(a) The proposed method (b) Distance greedy method (c) Stochastic distributed method

Fig. 8. The QoE under different methods.



(a) The proposed method (b) Distance greedy method (c) Stochastic distributed method

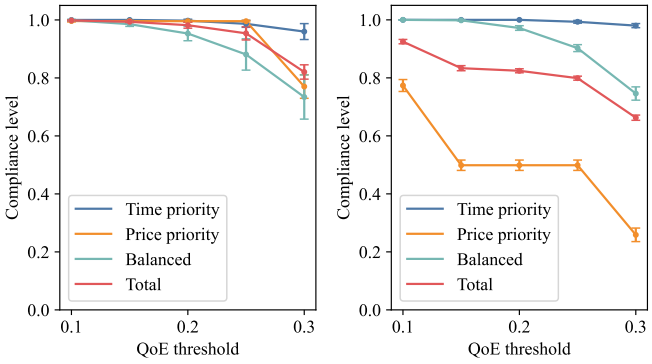
Fig. 9. The average charging price and average time cost under different methods.

in [19]. The results are illustrated in Fig. 8. In these figures, the color of the cells represents the average QoE across all EVs.

Comparing Fig. 8a Fig. 8b and 8c, it can be observed that our method consistently achieves higher QoE for all combinations of recommendation mode proportions, while the distance greedy method and the stochastic distributed method result in relatively lower QoE. The average QoE under all different mode combinations are 0.415, 0.165 and 0.330, respectively. This demonstrates the superiority of our method.

To further illustrate the performance in improving both time and cost aspects, Fig. 9 depicts the average charging price and average time cost. The color of the cells represents the normalized average time cost, while the size of the cells represents the normalized average charging price. It can be concluded from Fig. 9 that compared to the distance greedy method, our method consistently results in smaller average charging prices and smaller average time costs for all combinations of recommendation mode proportions. Despite the stochastic distributed method gaining an advantage in terms of average time cost, the higher average charging price makes it perform less effectively in QoE compared to our method.

5) Effectiveness validation of offering three modes: To illustrate the advantages of providing three optional recommendation modes compared to recommendations based solely on time-related metrics, we assume that when the QoE exceeds a threshold, the driver follows the recommendation; otherwise, the driver rejects the recommendation. We test the recommendation compliance level of our proposed method and time greedy method in ten random scenarios under different QoE thresholds. The results are shown in Fig. 10. Although in the time greedy method, users have only one choice, we assume that users still have different preferences. Different colored



(a) The proposed method (b) Distance greedy method

Fig. 10. The compliance level under different methods.

lines in the figure represent the compliance levels of different preference groups as well as the overall compliance level of all EVs. Each point on the line represents the average compliance level, and the error bars reflect the standard deviation.

From Fig. 10, it can be found that for the groups opting for the time priority and balanced modes, the compliance levels of the two methods are similar. However, since our method takes into account the needs of the group sensitive to charging price, it can significantly improve the compliance level for this group. In Fig. 10, the average total compliance level under five different QoE thresholds are 0.950 and 0.809, respectively. Our method has increased the overall compliance level by 17.4%.

6) Impact on PV consumption and charging price: To validate the effectiveness of our method in promoting PV consumption and reducing charging price, we fine-tune the

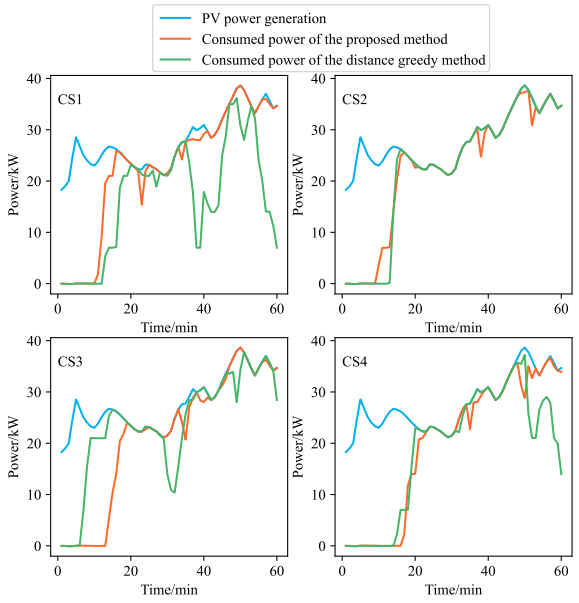


Fig. 11. The power generated by PV panels and the consumed power at each CS under different methods.

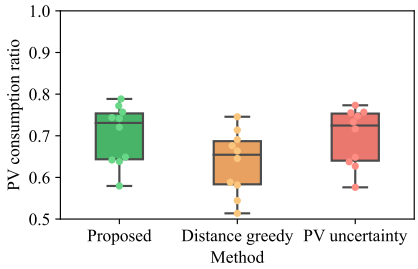


Fig. 12. PV consumption ratio distribution under 10 scenarios.

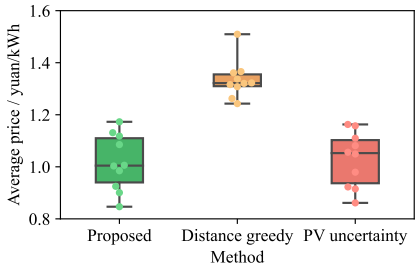


Fig. 13. Average charging price under 10 scenarios.

model in the same scenario as V-B1, except for $\omega^{\mathrm{P}} = 1$. Then, the fine-tuned model is deployed to another scenario where the vehicle insertion rate is 5, and the PV generation data is from August 16th, 10:00-11:00. We depict the curves of the consumed power and PV generation power at each CS under our method and under distance greedy method in Fig. 11.

Based on Fig. 11, the method of distance greedy results in a significant gap between the power generated by PVs and the actual consumed power at CS1 and CS4. Meanwhile, the power generated by PVs at CS2 is completely consumed for a prolonged period. This indicates that the vehicles charging at CS2 have reached capacity saturation, and are not redirected to other CSs to aid in the PV consumption.

In contrast, when the power generated by PV is not fully consumed, our method attempts to recommend vehicles to CSs with available PV power, thereby facilitating PV consumption. Consequently, these surplus power levels are promptly regulated.

To encourage EVs to participate in the consumption of PV power, we assume that when there is surplus PV power, the CS offers a 50% discount on the charging price to attract EV drivers. This way, the charging recommendation can help EV drivers reduce charging costs. We further evaluate our method in ten different scenarios. The result is shown in Fig. 12 and Fig. 13. The average PV consumption ratios over ten scenarios are 0.703, 0.636 and 0.697, respectively. Our method enhances the PV consumption ratio by 10.5%.

From Fig. 12 and Fig. 13, it is evident that across various testing scenarios, our method not only outperforms in promoting PV consumption but also excels in reducing charging costs. Thus, it demonstrates the capability to effectively handle a range of situations.

The typical PV power generation data is obtained through forecasting, and due to the uncertainty in production, there exists a forecasting error. In [43], the error is considered to follow a normal distribution, and the existing PV forecasting algorithms [44] have already been able to keep the forecasting error within a very small range. To study the performance of our method under production uncertainty, we assume that the input PV power generation data to our method is based on the true value with an added error, where the error follows a distribution of $\mathcal{N} \sim (0, 0.03 * \max\{p^{\mathrm{gen}}\})$. From Fig. 12 and Fig. 13, it can be seen that production uncertainty has only a slight impact on the PV consumption ratio and the average charging price. Considering the production uncertainty, our method remains effective.

VI. Conclusion

A. Summary

This paper presents a tailored deep reinforcement learning method to address the personalized charging recommendation problem, while also accounting for PV consumption at CSs. The main contents of our work are as follows:

- We offer three distinct modes to EV drivers, catering to their individual preferences for charging price and time cost.
- We investigate the potential of employing charging recommendation to facilitate PV consumption at CSs.
- We design a transfer reinforcement learning method that incorporates λ-returns, which enhances learning efficiency and adaptability to different scenarios.
- Simulation results show that our method is effective in improving QoE, thereby promoting the recommendation compliance level by 17.4%. Additionally, it also enhances the PV consumption ratio by 10.5%.

B. Future Lines of Research

Before deploying our method in real-world applications, several improvements and considerations must be addressed. These aspects will be the focus of our future work, enhancing the robustness and applicability of our proposed method. These include devising a decentralized framework to accommodate to larger-scale CS deployments, enhancing the prediction method for surplus PV power to improve the PV consumption ratio, and addressing real-world complexities such as EV non-compliance with recommendations, varying charging durations, and the provision of diverse services like fast charging and battery swapping.

C. Advantages & Limitations

We summarize the advantages and limitations of the proposed method as follows:

Advantages:

- Our method does not require an explicit model to describe the dynamic environment, and when applied, it only requires the forward propagation of the neural network, with a computational speed at the millisecond level.
- Our method integrates λ-return, which can handle delayed rewards in charging recommendation problems.
- Due to the integration of transfer learning, our method has strong adaptability to different scenarios.
- Experimental results verify the effectiveness of our method in enhancing QoE and promoting photovoltaic consumption.

Limitations:

- Reinforcement learning learns from trial-and-error interactions, which means that during the training phase, the experience of some EVs that are randomly recommended CSs will deteriorate, but this can be mitigated by sim-to-real transfer during actual deployment.
- The method we propose is centralized, and there will be service delays when a large number of vehicles make requests simultaneously.

REFERENCES

[1] China NEV sales to hit 8M in 2023, growth expected to slow. [Online]. Available: https://evmarketsreports.com/china-nev-sales-to-hit-8m-in-2023-growth-expected-to-slow/

[2] Z. Ye, Y. Gao, and N. Yu, "Learning to operate an electric vehicle charging station considering vehicle-grid integration," *IEEE Trans. Smart Grid*, vol. 13, no. 4, pp. 3038–3048, Mar. 2022.

[3] Z. Zhang, Y. Wan, J. Qin, W. Fu, and Y. Kang, "A Deep RL-Based Algorithm for Coordinated Charging of Electric Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 18 774–18 784, May. 2022.

[4] F. Kuipers, R. Kooij, D. De Vleeschauwer, and K. Brunnström, "Techniques for measuring quality of experience," in *Lect. Notes Comput. Sci.*, E. Osipov, A. Kassler, T. M. Bohnert, and X. Masip-Bruin, Eds., Berlin, Heidelberg, 2010, pp. 216–227.

[5] P. Alaee, J. Bems, and A. Anvari-Moghaddam, "A review of the latest trends in technical and economic aspects of ev charging management," *Energies*, vol. 16, no. 9, Apr. 2023.

[6] Y. Cao, H. Wang, D. Li, and G. Zhang, "Smart online charging algorithm for electric vehicles via customized actor-critic learning," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 684–694, Jan. 2022.

[7] Z. Wan, H. Li, H. He, and D. Prokhorov, "Model-free real-time EV charging scheduling based on deep reinforcement learning," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5246–5257, Sep. 2019.

[8] L. Yao, W. H. Lim, and T. S. Tsai, "A real-time charging scheme for demand response in electric vehicle parking station," *IEEE Trans. Smart Grid*, vol. 8, no. 1, pp. 52–62, Jan. 2017.

[9] M. Pourmatin, A. Fayaz-Heidari, M. Moeini-Aghtaie, E. Hassannayebi, and M. Basirati, "Investigating the sustainable development of charging stations for plug-in electric vehicles: A system dynamics approach," in *IFIP Advances in Information and Communication Technology*, Trondheim, Norway, 2023, pp. 400–416.

[10] M. Gharbaoui, L. Valcarenghi, R. Brunoi, B. Martini, M. Conti, and P. Castoldi, "An advanced smart management system for electric vehicle recharge," in *IEEE Int. Electr. Veh. Conf., IEVC*, Greenville, SC, United states, Mar. 2012, pp. 1–8.

[11] S. Liu, X. Xia, Y. Cao, Q. Ni, X. Zhang, and L. Xu, "Reservation-Based EV charging recommendation concerning charging urgency policy," *Sustainable Cities Soc.*, vol. 74, Nov. 2021.

[12] Y. Cao, O. Kaiwartya, Y. Zhuang, N. Ahmad, Y. Sun, and J. Lloret, "A decentralized deadline-driven electric vehicle charging recommendation," *IEEE Syst. J.*, vol. 13, no. 3, pp. 3410–3421, Sep. 2019.

[13] F. Elghitani and E. F. El-Saadany, "Efficient assignment of electric vehicles to charging stations," *IEEE Trans. Smart Grid*, vol. 12, no. 1, pp. 761–773, Jan. 2021.

[14] Z. Tian, T. Jung, Y. Wang, F. Zhang, L. Tu, C. Xu, C. Tian, and X.-Y. Li, "Real-time charging station recommendation system for electric-vehicle taxis," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 11, pp. 3098–3109, Apr. 2016.

[15] M. Aljaidi, N. Aslam, X. Chen, O. Kaiwartya, Y. A. Al-Gumaei, M. Khalid, and Ieee, "A reinforcement learning-based assignment scheme for EVs to charging stations," in *IEEE Veh Technol Conf*, Helsinki, Finland, Jun. 2022.

[16] S. R. Etesami, W. Saad, N. Mandayam, and H. V. Poor, "Smart routing in smart grids," in *IEEE Annu. Conf. Decis. Control, CDC*, Melbourne, VIC, Australia, Dec. 2017, pp. 2599–2604.

[17] W. Zhang, H. Liu, F. Wang, T. Xu, H. Xin, D. Dou, and H. Xiong, "Intelligent electric vehicle charging recommendation based on multi-agent reinforcement learning," in *Web Conf. - Proc. World Wide Web Conf., WWW*, Ljubljana, Slovenia, Apr. 2021, pp. 1856–1867.

[18] E. S. Rigas, S. D. Ramchurn, N. Bassiliades, and G. Koutitas, "Congestion management for urban EV charging systems," in *IEEE Int. Conf. Smart Grid Commun. SmartGridComm*, Vancouver, BC, Canada, Oct. 2013, pp. 121–126.

[19] M. Moschella, P. Ferraro, E. Crisostomi, and R. Shorten, "Decentralized assignment of electric vehicles at charging stations based on personalized cost functions and distributed ledger technologies," *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11 112–11 122, Jul. 2021.

[20] U. Fretzen, M. Ansarin, and T. Brandt, "Temporal city-scale matching of solar photovoltaic generation and electric vehicle charging," *Appl. Energy*, vol. 282, Jan. 2021.

[21] M. S. Islam and N. Mithulananthan, "PV based EV charging at universities using supplied historical PV output ramp," *Renewable Energy*, vol. 118, pp. 306–327, Apr. 2018.

[22] G. R. C. Mouli, M. Kefayati, R. Baldick, and P. Bauer, "Integrated PV charging of EV fleet based on energy prices, V2G, and offer of reserves," *IEEE Trans. Smart Grid*, vol. 10, no. 2, pp. 1313–1325, Oct. 2019.

[23] G. Ram, C. Mouli, and P. Bauer, "Optimal system design for a solar powered EV charging station," in *IEEE Transp. Electr. Conf. Expo, ITEC*, Long Beach, CA, United states, 2018, pp. 1094–1099.

[24] P. Xu, J. Zhang, T. Gao, S. Chen, X. Wang, H. Jiang, and W. Gao, "Real-time fast charging station recommendation for electric vehicles in coupled power-transportation networks: A graph reinforcement learning method," *Int. J. Electr. Power Energy Syst.*, vol. 141, Oct. 2022.

[25] H. Lin, X. Lin, H. Labiod, and L. Chen, "Toward multiple-phase mdp model for charging station recommendation," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 10 583–10 595, Aug. 2022.

[26] K.-B. Lee, M. A. Ahmed, D.-K. Kang, and Y.-C. Kim, "Deep reinforcement learning based optimal route and charging station selection," *Energies*, vol. 13, no. 23, Dec. 2020.

[27] X. Chen, W. Wei, Q. Yan, N. Yang, and J. Huang, "Time-delay deep q-network based retarder torque tracking control framework for heavy-duty vehicles," *IEEE Trans. Veh. Technol.*, vol. 72, no. 1, pp. 149–161, 2023.

[28] S. Z. Gou and Y. Liu, "Dqn with model-based exploration: efficient learning on environments with sparse rewards," 2019.

[29] J. A. Arjona-Medina, M. Gillhofer, M. Widrich, T. Unterthiner, J. Brandstetter, and S. Hochreiter, "Rudder: Return decomposition for delayed rewards," in *Adv. neural inf. proces. syst.*, vol. 32, Vancouver, BC, Canada, 2019.

[30] Y. Bouteiller, S. Ramstedt, G. Beltrame, C. Pal, and J. Binas, "Reinforcement learning with random delays," in *ICLR - Int. Conf. Learn. Represent*, Virtual, Online, 2021.
[31] B. Daley and C. Amato, "Reconciling λ-returns with experience replay," in *Adv. neural inf. proces. syst.*, vol. 32, Vancouver, BC, Canada, Feb. 2019.
[32] Z. Wang, C. Chen, and D. Dong, "Lifelong incremental reinforcement learning with online bayesian inference," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 8, pp. 4003–4016, Aug 2022.
[33] Y. Liu, C. Shu, J. Wang, and C. Shen, "Structured knowledge distillation for dense prediction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 6, pp. 7035–7049, Jun 2023.
[34] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, 2021.
[35] Z. Zhu, K. Lin, A. K. Jain, and J. Zhou, "Transfer learning in deep reinforcement learning: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 13 344–13 362, 2023.
[36] Z. Wan, H. Li, H. He, and D. Prokhorov, "Model-free real-time ev charging scheduling based on deep reinforcement learning," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5246–5257, Sep. 2019.
[37] Z. Ma, S. Zou, and X. Liu, "A distributed charging coordination for large-scale plug-in electric vehicles considering battery degradation cost," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 5, pp. 2044–2052, Sep. 2015.
[38] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *Int. Conf. Mach. Learn., ICML*, New York City, NY, United states, 2016, pp. 1995–2003.
[39] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," in *Int. Conf. Learn. Represent., ICLR - Conf. Track Proc.*, San Juan, Puerto rico, 2016.
[40] P. Li, M. Wei, H. Ji, W. Xi, H. Yu, J. Wu, H. Yao, and J. Chen, "Deep reinforcement learning-based adaptive voltage control of active distribution networks with multi-terminal soft open point," *Int. J. Electr. Power Energy Syst.*, vol. 141, Oct. 2022.
[41] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *IEEE Conf Intell Transport Syst Proc ITSC*, Maui, HI, United states, Nov. 2018, pp. 2575–2582.
[42] Photovoltaic (PV) Solar Panel Energy Generation data. [Online]. Available: https://data.london.gov.uk/dataset/photovoltaic--pv--solar-panel-energy-generation-data?q=solar%20pane
[43] S. Chang, Y. Niu, and T. Jia, "Coordinate scheduling of electric vehicles in charging stations supported by microgrids," *Electr. Power Syst. Res.*, vol. 199, Oct. 2021.
[44] Z. Si, M. Yang, Y. Yu, and T. Ding, "Photovoltaic power forecast based on satellite images considering effects of solar position," *Applied Energy*, vol. 302, Nov. 2021.

Wenlei Chen was born in Fujian, China, in 2000. He received the B.Sc. degree in automation from Central South University, Changsha, China, in 2022. He is currently pursuing the master's degree with the Department of Automation, Tsinghua University, Beijing.

His current research focuses on renewable energy conversion and reinforcement learning, specifically in the context of energy Internet.

Di Liu was born in Henan, China, in 1990. He is currently an assistant researcher in the Department of Electrical Engineering at Tsinghua University. He received the B.S degree in electrical engineering and management in 2013, the M.S degree in electronic and communication engineering in 2015, and the Ph.D. degree in electrical engineering in 2020, all from North China Electric Power University, Beijing, China.

He conducted postdoctoral research at Tsinghua University from 2020 to 2024. His research interests include demand side management, electricity market, and energy Internet.

Junwei Cao (Senior Member, IEEE) received the bachelor's and master's degrees in control theories and engineering from Tsinghua University, Beijing, China, in 1998 and 1996, respectively, and the Ph.D. degree in computer science from the University of Warwick, Coventry, U.K., in 2001.

He is currently a Professor and a Vice Dean of Research Institute of Information Technology, Tsinghua University. He is also the Director of Open Platform and Technology Division, Tsinghua National Laboratory for Information Science and Technology. Prior to joining Tsinghua University in 2006, he was a Research Scientist with MIT LIGO Laboratory and NEC Laboratories Europe for about five years. He has published over 200 papers and cited by international scholars for over 18 000 times. He has authored or edited eight books. His research is focused on distributed computing technologies and energy/power applications. He is a Senior Member of the IEEE Computer Society and a member of the ACM and CCF.