Contents lists available at ScienceDirect

Renewable and Sustainable Energy Reviews

journal homepage: www.elsevier.com/locate/rser

Reinforcement learning for electric vehicle applications in power systems: A critical review

Dawei Qiu [a], Yi Wang [a,*], Weiqi Hua [b], Goran Strbac [a]

[a] Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, UK
[b] Department of Engineering Science, University of Oxford, OX1 3QG, UK

ARTICLE INFO

Keywords:
Electric vehicles
Vehicle-to-grid
Reinforcement learning
Power systems

ABSTRACT

Electric vehicles (EVs) are playing an important role in power systems due to their significant mobility and flexibility features. Nowadays, the increasing penetration of renewable energy resources has been observed in modern power systems, which brings many benefits for improving climate change and accelerating the low-carbon transition. However, the intermittent and unstable nature of renewable energy sources introduces new challenges to both the planning and operation of power systems. To address these issues, vehicle-to-grid (V2G) technology has been gradually recognized as a valid solution to provide various ancillary service provisions for power systems. Many studies have developed model-based optimization methods for EV dispatch problems. Nevertheless, this type of method cannot effectively handle the highly dynamic and stochastic environment due to the complexity of power systems. Reinforcement learning (RL), a model-free and online learning method, can capture various uncertainties through numerous interactions with the environment and adapt to various state conditions in real-time. As a result, using advanced RL algorithms to solve various EV dispatch problems has attracted a surge of attention in recent years, leading to many outstanding research papers and important findings. This paper provides a comprehensive review of popular RL algorithms categorized by single-agent RL and multi-agent RL, and summarizes how these advanced algorithms can be applied to various EV dispatch problems, including grid-to-vehicle (G2V), vehicle-to-home (V2H), and V2G. Finally, key challenges and important future research directions are discussed, which involve five aspects: (a) data quality and availability; (b) environment setup; (c) safety and robustness; (d) training performance; and (e) real-world deployment.

1. Introduction

Over the last decades, modern power systems have undergone major changes in various aspects due to a number of technical, economic, and environmental factors. One of the most remarkable things is associated with climate change, which has altered our energy policy and energy mix to a low-carbon transition [1,2]. The Committee on Climate Change (CCC), the UK's independent climate advisory body, claims that by setting an ambitious new target to reduce greenhouse gas emissions to net zero by 2050, the UK can halt its contribution to global warming within 30 years [3]. To achieve this target, large penetration of renewable energy resources (RESs) into power systems has been witnessed in recent years. Nevertheless, their intermittent nature leads to new challenges to system stability and security [4]. As one of the most important demand-side technologies, electric vehicles (EVs) can provide various ancillary services for stable and secure power system operations via vehicle-to-home (V2H) and vehicle-to-grid (V2G) technology [5], thereby contributing to the increasing integration of EVs in modern power systems.

Integrating large-scale EVs into power systems can benefit the low-carbon transition and system stability, but also introduces challenges to effective EV dispatch due to the potential privacy concerns and the difficulty of handling various system uncertainties and dynamics [6]. In recent years, model-based optimization methods have been widely developed to model the EV dispatch problems for reliable traveling behaviors through local charging stations and various ancillary services through V2G technologies, e.g., energy imbalance service, carbon intensity service, voltage support, frequency regulation, etc. However, the limitations of model-based optimization methods cannot be erased and have been listed as follows [7,8]:

- Model-based optimization methods assume that EVs require complete knowledge of the experiment environment, e.g., power network, transport status, uncertainty probability distribution, etc.

[*] Corresponding author.
E-mail address: yi.wang18@imperial.ac.uk (Y. Wang).

https://doi.org/10.1016/j.rser.2022.113052
Received 1 June 2022; Received in revised form 30 October 2022; Accepted 12 November 2022
Available online 29 November 2022

Nomenclature

Abbreviations

RL	Reinforcement learning
CCC	Committee on Climate Change
EV	Electric vehicle
G2V	Grid-to-vehicle
V2H	Vehicle-to-home
V2G	Vehicle-to-grid
RES	Renewable energy resource
DER	Distributed energy resource
HILP	High-impact and low-probability
AMI	Advanced metering infrastructure
MDP	Markov Decision Process
SoC	State-of-the-charge
DQN	Deep Q-network
DNN	Deep neural network
SARSA	State–action–reward–state–action
PPO	Proximal policy optimization
DDPG	Deep deterministic policy gradient
TD3	Twin delayed DDPG
SAC	Soft actor–critic
TD	Temporal difference
PG	Policy gradient
FRL	Federated reinforcement learning
CTCE	Centralized training with centralized execution
DTDE	Decentralized training with decentralized execution
CTDE	Centralized training with decentralized execution
PS	Parameter sharing
RNN	Recurrent neural network
LSTM	Long short-term memory
GRU	Gated recurrent unit
GNN	Graph neural network

Reinforcement learning

s	State
a	Action
r	Reward
α	Learning rate
ν	Soft updating rate
γ	Reward discount factor, $\gamma \in [0,1)$
$\mathcal{S}$	State space
$\mathcal{A}$	Action space
τ	Trajectory, which is a sequence of states, actions and rewards
$\mathcal{R}$	Replay buffer of storing experiences of states, actions and rewards
$\pi(a \mid s)$	Stochastic policy function, probability of taking action a in state s
$\mu(s)$	Deterministic policy function, action taken in state s
$p(s' \mid s,a)$	State-transition function, probability of transitioning to state s' from state s taking action a
$Q^{\pi}(s,a)$	Action-value function, value of taking action a in state s under policy π
$V^{\pi}(s)$	State-value function, value of state s under policy π
$\pi_{\phi}(a \mid s)$	Stochastic policy network parameterized by ϕ
$\mu_{\phi}(s)$	Deterministic policy network parameterized by ϕ
$Q_{\theta}(s,a)$	Action-value network parameterized by θ
$V_{\theta}(s,a)$	State-value network parameterized by θ
$\hat{A}$	Generalized advantage function

Electric vehicle

$t \in T$	Index and set of time steps
$i \in I$	Index and set of EVs
Δt	Time resolution
λ^{g}_{t}	Grid electricity price at time step t (£/kWh)
$\overline{P}_i$	Power capacity of EV i (kW)
$\overline{E}_i$	Energy capacity of EV i (kWh)
$\overline{S}_i$	Maximum battery SoC of EV i (%)
$E^{tp}_{i,t}$	Energy requirement for traveling of EV i during the horizon of time step t (kWh)
η^{c}_{i}	Charging efficiency of EV i (%)
η^{d}_{i}	Discharging efficiency of EV i (%)
$P^{c}_{i,t}$	Charging power of EV i at time step t (kW)
$P^{d}_{i,t}$	Discharging power of EV i at time step t (kW)
$E_{i,t}$	Battery energy content of EV i at time step t (kWh)
$S_{i,t}$	Battery SoC of EV i at time step t (%)
$A_{i,t}$	Binary indicating whether EV i connects with grid ($A_{i,t}=1$) or not ($A_{i,t}=0$) at time step t

However, such assumptions are normally impractical considering the highly stochastic and dynamic real-world environment.

- Model-based optimization methods normally handle uncertainties via stochastic programming or robust optimization, which may only be able to capture a small number of representative scenarios or lead to very conservative optimization results. Meanwhile, stochastic programming can be time-consuming, especially when a large number of scenarios are involved. Finally, the solutions need to be re-optimized in any new state.

Reinforcement learning (RL) [9] is regarded as a model-free method to study the sequential and dynamic decision-making problems of agents that can gradually learn the optimal control decisions by utilizing experiences acquired from their repeated interactions with the environment, without a *prior* knowledge. In addition, RL as an online learning method can make efficient use of increasing data from the environment, thereby capturing the system uncertainties and adapting to various state dynamics. Finally, once the RL algorithm is well trained, its policy can be directly delivered to the real-world new test set on timescales of milliseconds without requiring any identification. Therefore, RL is claimed as an efficient tool for real-time automatic control applications of EV dispatch problems. Specifically, the characteristics of both the model-based optimization method and the model-free RL method are illustrated in Fig. 1.

Meanwhile, the critical review of RL applications in the power and energy community has attracted surging attention, such as power systems [10], demand response [11], sustainable energy and electric

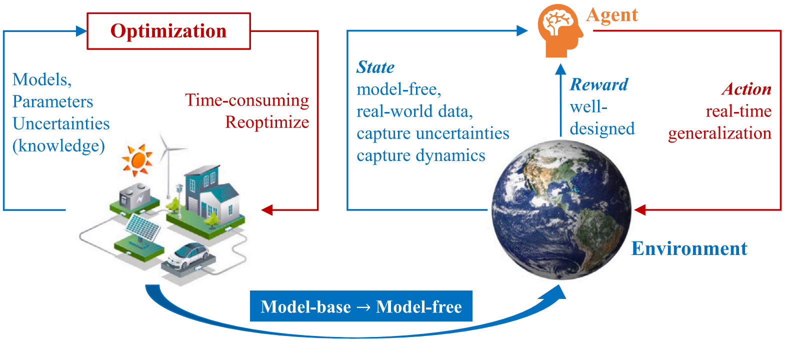


Fig. 1. Transition from model-based optimization method to model-free reinforcement-learning method.

systems [12,13], and building energy management [14,15]. RL-based decision-making methods are envisioned to compensate for the limitations of existing model-based optimization methods and thus are promising to address these emerging challenges. Although there are several review articles with regards to the EV concept, they do not focus on the dispatch problems from a data-driven learning approach, specifically with the application of RL algorithms. For instance, the review in [16] explains the current regulations, standards, and interfacing issues of EV transport electrification within the smart grid concept, such as power quality, reliability, and control. Various computational scheduling methods (including conventional mathematical optimization and meta-heuristic algorithm) for integrating EVs with power systems are reviewed and compared in [17]. Studies related to economic dispatch and risk management of large-scale EV-penetrated power systems in the electricity market are reviewed in [18]. The authors of [19] reviewed the digital twin technologies for the future smart EVs and analyzed the techno-socio-economic impact of digital twin technology on vehicle technology. In contrast, this paper provides a review of RL-based decision-making of EVs in power systems. We firstly introduce various RL algorithms, then exemplify how to apply RL to various EV dispatch problems, and finally discuss critical issues in their applications. Overall, the key contributions of this work are described as follows:

- Present a comprehensive and structural overview of RL algorithms in terms of basic concepts, theoretical fundamentals, and state-of-the-art RL algorithms, including both single-agent RL (SARL) and multi-agent RL (MARL) that have been applied to EV dispatch problems.
- Select three key applications of EV dispatch problems: grid-to-vehicle (G2V), vehicle-to-home (V2H), and vehicle-to-grid (V2G). Illustrate the overall procedure of applying RL algorithms to each of three applications in terms of modeling, solution, implementation, and discussion.
- Discuss the critical challenges and future perspectives for applying RL to EV dispatch problems in terms of data quality and availability, environment setup, safety and robustness, training performance, and real-world deployment.

Given the comprehensiveness and specificity of this review, it can assist the community in quickly locating relevant papers in this field and gaining a thorough understanding of current studies through the detailed comparison provided by this review. Additionally, this work can help both academia and industry comprehensively grasp research trends and challenges as well as clarify future research directions in the field of EV dispatch problems with RL techniques. Furthermore, the issues discussed in this review can guide policy makers and regulators with very useful insights in terms of energy, emissions/environment, costs, climate change, etc.

The rest of this paper is organized as follows. Section 2 describes the preliminaries of RL, including Markov decision process, the state-of-the-art SARL and MARL algorithms. Section 3 provides a comprehensive review of RL applications to G2V, V2H, and V2G. Section 4 discusses the key challenges and several potential future directions of RL-based EV dispatch problems. Finally, the conclusion of this work is drawn in Section 5.

2. Preliminaries of reinforcement learning

2.1. Markov decision process

The mathematical foundation of RL is the Markov Decision Process (MDP). An MDP usually consists of a state space, an action space, a state transition function, a reward function, and a discount factor. In general, RL is a sequential decision-making process that tries to find a decision rule (i.e., policy) that makes the studied entity obtain the maximum cumulative reward, i.e., get the maximum benefit. In order to facilitate the readers' understanding and memory, the following context mainly uses the example of an EV energy arbitrage problem [20] to explain the elements of an MDP.

- The agent i is the decision-maker in the problem. In the case of the EV energy arbitrage problem, an EV is defined as an agent that can perform charging and discharging behaviors.
- Environment $\mathcal{E}$ is the operation model of the problem. In the case of the EV energy arbitrage problem, the environment is defined as the energy management system model of the EV battery that can generate a new state.
- State $s \in \mathcal{S}$ is a description of the environment. In the case of the EV energy arbitrage problem, the price signals and/or battery state-of-the-charge (SoC) can be regarded as the state. The EV only needs to know its current state to make decisions and decide whether to charge or discharge. Specifically, the state can be realized as the only basis for making decisions. The state space $\mathcal{S}$ is then the collection set of all possible states, which can be either finite or infinite. In the case of the EV energy arbitrage, the state spaces of both price signals and battery SoC are infinite sets, since they are continuous values.
- Action $a \in \mathcal{A}$ is the decision made by the agent. In the case of the EV energy arbitrage, the action is the battery power schedule, e.g., charging from the main grid or discharging back to the main grid. The action space $\mathcal{A}$ is the collection set of all possible actions. In the case of the EV energy arbitrage, if the EV can only behave in the status of charging or discharging, then the action space is a discrete (finite) set $\mathcal{A} = \{\text{charging}, \text{discharging}\}$. If the EV can decide the exact value of battery charge or discharge power, then the action space is a continuous (infinite) set $\mathcal{A} = [-1, 1]$ that represents the magnitude of charging (positive) and discharging (negative) power as a percentage of its battery power capacity.
- The policy function $\pi(a|s)$ is used to generate the action in observing the state. In the case of the EV energy arbitrage, when the price and the battery SoC are both low, there is a high probability

that the EV agent will decide to charge its battery so as to save the energy cost but also ensure sufficient SoC for traveling.
The policy function can be defined in different ways. We introduce the most common one $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ as a conditional probability density function:

$$\pi(a|s) = \mathbb{P}(A = a|S = s), \tag{1}$$

where its input is the state s and its output is a probability value between 0 and 1 indicating the probability of selecting the action a. In the case of the EV energy arbitrage, take the state s as input to the policy function, its outputs the probability values of two actions:

$$\pi(\text{charge}|s) = 0.7, \tag{2}$$

$$\pi(\text{discharge}|s) = 0.3, \tag{3}$$

which means charging power with a probability of 0.7 and discharging power with a probability of 0.3. It can be observed that both two actions are possible, but the probability of charging power is higher than that of discharging power. It should be noted that the goal of RL is to learn the policy function $\pi(a|s)$. As long as there is an optimal policy function, it can automatically control the EV agent to achieve the best performance.

- The reward $r \in \mathcal{R}$ is a value returned to the agent by the environment after the agent performs an action a in state s. In general, rewards are defined based on the studied problem itself, which can greatly affect the performance of an RL policy. In the case of the EV energy arbitrage, the most straightforward reward can be defined as the negative energy cost, i.e., the higher the negative energy cost is, the better the control policy is.
- State transition function $p(s'|s, a)$ is the function utilized by the environment to generate a new state s'. Given the current state s, the agent i executes an action a, and the environment returns the state s' in the next step. Thus, it is a mapping from the current state and the executed action to a new state $(s, a) \rightarrow s'$.
In general, the state transition function can be deterministic or stochastic. In the case of the EV energy arbitrage, the state transition function of the battery SoC $S_{i,t}$ of EV agent i from time step t to $t+1$ can be expressed as:

$$S_{i,t+1} = \begin{cases} S_{i,t} + (P^c_{i,t}\eta^c_i + P^d_{i,t}/\eta^d_i)\Delta t/\overline{E}_i & \text{if } A_{i,t} = 1 \\ S_{i,t} - E^{tp}_{i,t}/\overline{E}_i & \text{if } A_{i,t} = 0, \end{cases} \quad \forall i \in \mathcal{I},\ \forall t \in \mathcal{T}, \tag{4}$$

where $P^c_{i,t}$ and $P^d_{i,t}$ indicate the charging (positive) and discharging (negative) power, while η^c_i and η^d_i correspond to the charging and discharging efficiencies, respectively. $\overline{E}_i$ is the battery energy capacity. $E^{tp}_{i,t}$ refers to the energy consumption for traveling purpose, and the binary $A_{i,t}$ indicates whether the EV agent i is connected to the grid ($A_{i,t} = 1$) or not ($A_{i,t} = 0$) at time step t. Meanwhile, the state transition function may also be stochastic, which is characterized by the inherent variability and uncertainty of the environment. In the case of the EV energy arbitrage, this corresponds to the exogenous state features, e.g., grid price signals, EV traveling patterns, demand profiles, PV generation, etc. In real-world applications, it presents significant challenges to identify suitable probabilistic models that can fully capture such randomness since it is influenced by many exogenous factors, such as driving behaviors, energy usage behaviors, solar radiation, and pricing schemes of utility companies. RL, however, remedies this problem in a data-driven approach that does not rely on accurate models of the underlying uncertainties but learns their probability characteristics through the historic data or experience acquired from the environment via machine learning techniques.

After introducing the elements of an MDP, we go further into the interaction between agent and environment, as depicted in Fig. 2. The agent observes the state s_t of the environment, makes an action a_t, the action then changes the state of the environment according to the state transition function $p(s'|s, a)$, and the environment feeds back to the agent a reward r_t and a new state s_{t+1}. This process continues until the end of the episode (e.g., one trading day), then emits a trajectory:

$$\tau = s_1, a_1, r_1,\ \ s_2, a_2, r_2,\ \ s_3, a_3, r_3,\ \ \ldots\ , s_T, a_T, r_T, \tag{5}$$

where T is the time horizon of the episode (e.g., 24 h). The objective of RL is to optimize a policy to maximize the expected return given all possible trajectories under the optimized policy. Mathematically, given a state distribution ρ and a policy π, the probability of the occurrence of a T-step trajectory in an MDP can be expressed as:

$$p(\tau|\pi) = \rho(s_1)\prod_{t=1}^{T} p(s_{t+1}|s_t, a_t)\pi(a_t|s_t). \tag{6}$$

Given the reward r and all possible trajectories τ, the expected reward $J(\pi)$ can be defined as:

$$J(\pi) = \int_{\tau} p^{\tau}(\tau|\pi)r(\tau) = \mathbb{E}_{\tau\sim\pi}[r(\tau)], \tag{7}$$

where p^{τ} represents the probability of trajectory occurrence, and the higher the probability of occurrence, the greater the weight of the expected return calculation. The RL improves the policy by optimization methods to maximize the expected returns. The optimal policy π^* can be expressed as:

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, J(\pi). \tag{8}$$

In the MDP, given a state s, there is a state-value function $V(s)$ that represents the expected return using policy π, which can be defined as:

$$\begin{aligned} V^{\pi}(s) &= \mathbb{E}_{\tau\sim\pi}[r(\tau)|s_1 = s] \\ &= \mathbb{E}_{a_t\sim\pi(\cdot|s_t)}[\sum_{t=1}^{T}\gamma^t r(s_t, a_t)|s_1 = s], \end{aligned} \tag{9}$$

where $\tau \sim \pi$ indicates the samples of trajectory τ that are obtained by policy π, while $a_t \sim \pi(\cdot|s_t)$ represents the action a_t sampled from the policy π in observing state s_t, and $r(s_t, a_t)$ is the reward calculated given the current state s_t and action a_t. Finally, $\gamma \in [0, 1)$ is the discount factor to expect the long-term return of the trajectory τ.

Furthermore, given an action a, there is an action-value function (or Q-value function), which depends on the current state and the action just performed, and is related to the expected return. If an agent follows policy π, the action-value function is defined as $Q^{\pi}(s, a)$ and written as:

$$\begin{aligned} Q^{\pi}(s, a) &= \mathbb{E}_{\tau\sim\pi}[r(\tau)|s_1 = s, a_1 = a] \\ &= \mathbb{E}_{a_t\sim\pi(\cdot|s_t)}[\sum_{t=1}^{T}\gamma^t r(s_t, a_t)|s_1 = s, a_1 = a]. \end{aligned} \tag{10}$$

Finally, the relationship between the state-value function $V^{\pi}(s)$ and action-value function $Q^{\pi}(s, a)$ can be expressed as:

$$V^{\pi}(s, a) = \mathbb{E}_{a\sim\pi}[Q^{\pi}(s, a)] \tag{11}$$

2.2. Single-agent reinforcement learning

After reviewing the research work that applies RL to EV dispatch problems, we can theoretically provide a broad classification of model-free RL algorithms by categorizing them into two sets: value-based and policy-based (as depicted in Fig. 3). In the value-based set, Q-learning [21] is the classical RL algorithm for learning action-value function, while deep Q-network (DQN) [22] is the originator of the deep RL (DRL) algorithm that approximates the action-value function via deep neural networks (DNN). In addition, state–action–reward–state–action (SARSA) [9] and fitted Q-iteration [23] are also involved.

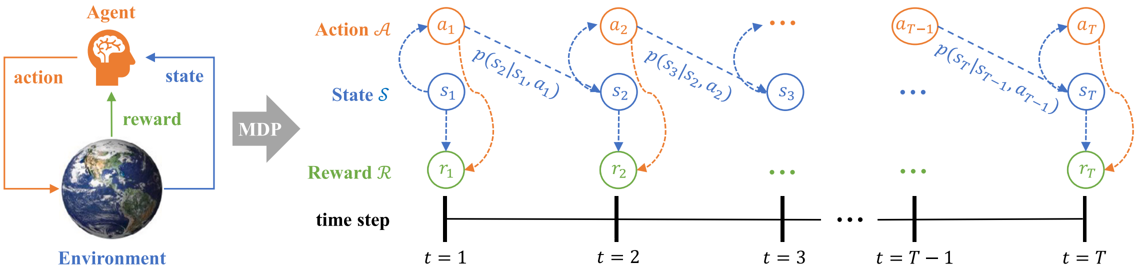


Fig. 2. Agent–environment interaction and the process of Markov decision process.

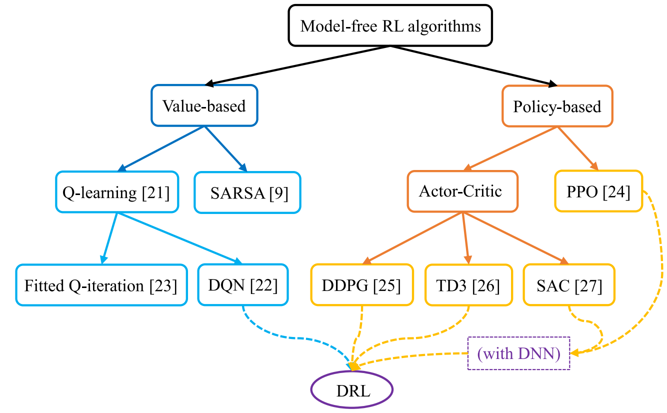


Fig. 3. Classifications of single-agent reinforcement learning algorithms.

Additionally, policy-based RL algorithms mainly include the policy gradient theorem of proximal policy optimization (PPO) [24], and the actor–critic methods of deep deterministic policy gradient (DDPG) [25], twin delayed DDPG (TD3) [26], and soft actor–critic (SAC) [27]. Theoretically, DQN, DDPG, and TD3 all belong to the DRL algorithm. However, both PPO and SAC can also be categorized into the DRL algorithm if they employ DNNs as function approximators as well.

2.2.1. Q-learning

Q-Learning [21] is a tabular approach based on temporal difference (TD) learning [28]. It is assumed that both state and action are in discrete spaces (e.g., $\mathcal{S}$ has 4 possible states and $\mathcal{A}$ has 3 potential actions), the optimal Q-value function $Q^*(s,a)$ can be represented as a 4 × 3 table. In observing the current state s_t, the optimal action can be selected as:

$$a_t = \operatorname*{argmax}_{a\in\mathcal{A}} Q^*(s_t,a) \tag{12}$$

that means finding the maximum Q-value of the row corresponding to state s_t, and returning the action associated with that Q-value. In Q-learning, a Q-table is used to approximate the Q-value function $Q^*(s,a)$. It first initializes all elements in the table to zero and then updates one element of the table at a time. Eventually, the Q-table will converge to the optimal $Q^*(s,a)$. In order to update this Q-table, it uses the optimal Bellman equation [9] and can write the updated Q-table as:

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha[\underbrace{\overbrace{r_t + \gamma \max_{a_{t+1}} Q(s_{t+1},a_{t+1})}^{\text{target Q}} - Q(s_t,a_t)}_{\text{TD error}}], \tag{13}$$

where α is the learning rate that governs the Q-table's updating steps. It is noted that when using a tabular approach, the Q-value function can be represented as a large two-dimensional table. That is, there is a separate entry for each discrete state and action. However, this approach will be very inefficient when solving the task with a large data space, suffering severely from the curse of dimensionality [22].

2.2.2. Deep Q-network (DQN)

In order to overcome the discrete state space in Q-learning, DQN [22] employs a deep neural network (DNN) (parameterized by θ) as a function approximator to represent the Q-value function in multi-dimensional continuous state space $Q(s_t,a_t) \approx Q_\theta(s_t,a_t)$. Specifically, DQN learns this Q-value function $Q_\theta(s_t,a_t)$ corresponding to the optimal policy by minimizing the loss (i.e., TD error):

$$\mathcal{L}(\theta) = \mathbb{E}_{s_t,a_t,r_t,s_{t+1}\sim\mathcal{R}}[(r_t + \gamma \max_{a_{t+1}} Q_{\theta'}(s_{t+1},a_{t+1}) - Q_\theta(s_t,a_t))^2], \tag{14}$$

where $Q_{\theta'}(\cdot)$ is a target Q-value function whose parameters are periodically updated with the most recent $Q_\theta(\cdot)$, which can help stabilize the training performance. Another crucial technique for stabilization in DQN is the use of an experience replay buffer $\mathcal{R}$, which contains tuples (experiences) (s_t,a_t,r_t,s_{t+1}) that the agent interacts with the environment. It is known that there is a correlation between two consecutive experiences because the state transition is continuous. If directly taking a batch of experiences in sequence as the training set, the policy is easy to overfit, since the training samples are not independent [29]. To solve this problem, we can randomly select a small number of experiences from the buffer $\mathcal{R}$ as a batch, which not only ensures that the training samples are independent and equally distributed, but also makes the size of each batch small enough to accelerate the training speed [22].

2.2.3. Policy gradient (PG)

PG is another popular choice for a variety of RL methods that is able to solve problems with continuous state space. PG [30] employs a DNN (parameterized by ϕ) that takes a continuous state s_t as input and outputs a stochastic policy $\pi_\phi(a|s)$ representing the selection probabilities of action a in an observed state s. The main idea is to directly adjust the parameters ϕ of the policy in order to maximize the objective $J(\pi) = \mathbb{E}_{s\sim\rho,a\sim\pi}[r(\tau)|s_1 = s, a_1 = a]$ by moving in the direction of $\nabla_\phi J(\pi)$, so that the log-probability of choosing actions proportionate to the sampled return $r(\tau)$ is increased. Using the Q-value function defined in DQN, the gradient of the policy can be written as [31]:

$$\nabla_\phi J(\pi) = \mathbb{E}_{s_t\sim\rho,a_t\sim\pi}\left[\nabla_\phi \log \pi_\phi(a_t|s_t) Q_\theta(s_t,a_t)\right], \tag{15}$$

where ρ is the state distribution. In addition, PG is capable of handling continuous control by representing the probability distribution of the agent's action with a Gaussian distribution $\mathcal{N}(\mu_t, \sigma_t^2)$, and predicting the mean $\mu_\phi(s_t)$ and the standard deviation $\sigma_\phi(s_t)$ of it with two DNNs, referring to a Gaussian Policy [31].

2.2.4. Proximal policy optimization (PPO)

PPO is an advanced PG algorithm that can achieve a balance between the ease of implementation, sampling efficiency, and ease of tuning [24]. In other words, training a relatively good performance in the vanilla PG algorithm is very challenging, since it is very sensitive to the learning rate, i.e., a small learning rate takes a long time to make the training converge, while a large learning rate easily falls into the local optimum. However, PPO can effectively address the difficulty by constructing a probability ratio between the new and old policies, and then clipping it within a stable interval. In this case, the policy of PPO can be updated in a trust region. Similar to many PG algorithms, PPO is applicable for modeling multi-dimensional continuous state and action spaces.

To model the action characteristics in continuous domain, we generate a set of Gaussian distributions for the policy network parameterized by ϕ to output the corresponding mean and standard deviation for all action dimensions, the stochastic policy $\pi_\phi(a|s)$ is then sampled for the optimal action a_t in state s_t. This stochastic policy can be updated by maximizing its clipped surrogate objective that considers the restriction of policy update:

$$\mathcal{L}_t^{\text{CLIP}}(\phi) = \mathbb{E}_t\left[\min(\zeta_t \hat{A}_t, \text{clip}(\zeta_t, 1-\epsilon, 1+\epsilon)\hat{A}_t)\right], \tag{16}$$

where the first term $\zeta_t \hat{A}_t$ within the operator $\min\{\cdot\}$ indicates the normal policy gradient, while the second term $\text{clip}(\zeta_t, 1-\epsilon, 1+\epsilon)\hat{A}_t$ within the operator $\min\{\cdot\}$ trims the policy gradient by clipping the probability ratio ζ_t^d between $[1-\epsilon, 1+\epsilon]$. The hyperparameter $\epsilon \in [0,1]$ is used to truncate the gradient update of the new policy from the old version. In other words, the advantage function $\hat{A}_t$ will be clipped if the probability ratio goes beyond the range $[1-\epsilon, 1+\epsilon]$. In PPO policy, the probability ratio ζ_t can be expressed as:

$$\zeta_t = \frac{\pi_\phi(a_t|s_t)}{\pi_{\phi\text{old}}(a_t|s_t)}, \tag{17}$$

In addition, the generalized advantage function $\hat{A}_t$ in (16) can be expressed as:

$$\hat{A}_t = \delta_t + \gamma\delta_{t+1} + \cdots + \gamma^{T-t+1}\delta_{T-1}, \tag{18}$$

$$\text{where } \delta_t = r_t + \gamma V_\psi(s_{t+1}) - V_\psi(s_t), \tag{19}$$

here $V_\psi(s)$ is the state-value function that can be approximated by a state-value network parameterized by ψ.

2.2.5. Deep deterministic policy gradient (DDPG)

DDPG is a successful DRL method to deal with high-dimensional and continuous state and action spaces that features an actor–critic architecture and employs two DNNs with different purposes [31], aiming to derive directly the deterministic policies $\mu_\phi : \mathcal{S} \rightarrow \mathcal{A}$. The actor network μ_ϕ takes as input a state s and implements the policy improvement task, updating the policy with respect to the estimated Q-value function and outputting a continuous action $a = \mu_\phi(s)$. The critic network Q_θ takes as input a state s and an action a and outputs a scalar estimate of the Q-value function $Q_\theta(s,a)$.

Similarly as DQN, DDPG also incorporates an experience replay buffer $\mathcal{R}$ that stores the past experiences (s_t, a_t, r_t, s_{t+1}) and samples a minibatch of experiences to update the networks. Furthermore, DDPG introduces target networks for both actor and critic, denoted as $\mu_{\phi'}(s_t)$ and $Q_{\theta'}(s_t, a_t)$, respectively, then adopts soft update that lies in restricting the target values to change slowly so as to stabilize the learning process, which can be expressed as $\phi' \leftarrow \nu\phi + (1-\nu)\phi'$ and $\theta' \leftarrow \nu\theta + (1-\nu)\theta'$ with the updating rate $\nu \ll 1$. By employing these two techniques of experience replay buffer and target networks, the parameters θ of the critic network can be optimized to minimize the TD error defined as:

$$\mathcal{L}(\theta) = \mathbb{E}_{s_t\sim\rho^\mu, a_t\sim\pi_\phi}[(Q_\theta(s_t,a_t) - y_t)^2], \tag{20}$$

where the target value is:

$$y_t = r_t + \gamma Q_{\theta'}(s_{t+1}, \mu_{\phi'}(s_{t+1})). \tag{21}$$

According to the deterministic policy gradient theorem [31], the parameters ϕ of the actor network can be optimized using the gradient ascent algorithm with the computed gradient defined as:

$$\nabla_\phi J(\mu_\phi) = \mathbb{E}_{s_t\sim\rho}[\nabla_\phi\mu_\phi(s_t)\nabla_{a_t}Q_\theta(s_t, \mu_\phi(s_t))]. \tag{22}$$

In addition, to aid the agent in thoroughly exploring the environment, we construct an exploration policy $\hat{\mu}(s_t) = \mu_\phi(s_t) + \mathcal{N}(0, \sigma_t^2 I)$ by adding a Gaussian noise $\mathcal{N}(0, \sigma_t^2 I)$ to the actor's output $\mu_\phi(s_t)$.

2.2.6. Soft actor critic (SAC)

The two main challenges of RL are high sample complexity and weak convergence. SAC [27] is proposed based on the maximum entropy framework to address these two challenges. Rather than just improving the policy by estimating the Q-value of the policy π, SAC further extends the soft policy iteration to a more practical function approximation setting. It learns by alternately optimizing between the value function and the policy function. Specifically, SAC includes a soft state-value function $V_\psi(s_t)$, two soft Q-value functions $Q_{\theta_1}(s_t,a_t)$ and $Q_{\theta_2}(s_t,a_t)$, and a policy function $\pi_\phi(a_t|s_t)$, whose parameters are represented as ψ, θ_1, θ_2 and ϕ, respectively. In which the value function can be directly modeled as a DNN, and the policy function is modeled as a Gaussian distribution, whose mean and standard deviation are estimated by the DNN. In general, we do not estimate the state-value function because it can be determined by the Q-value function and policy function, but in practice, including this term improves training stability and allows for easy co-training with other networks.

The soft Q-value functions $Q_{\theta_j}(s_t,a_t)$, where $j = \{1,2\}$ is updated by using the soft Bellman residual, which adds entropy term compared to Bellman residual as follows:

$$\mathcal{L}(\theta_j) = \mathbb{E}_{s_t,a_t,r_t,s_{t+1}\sim\mathcal{R}}[(Q_{\theta_j}(s_t,a_t) - y_t)^2], \forall j \in \{1,2\}, \tag{23}$$

where the target value is:

$$y_t = r_t + \gamma\mathbb{E}_{s_{t+1}\sim\rho}[V_\psi(s_{t+1})]. \tag{24}$$

The soft state-value function $V_\psi(s_t)$ is updated as:

$$\mathcal{L}(\psi) = \mathbb{E}_{s_t\sim\mathcal{R}}[(V_\psi(s_t) - \mathbb{E}_{a_t\sim\pi_\phi}[\min_{j=1,2} Q_{\theta_j}(s_t,a_t) - \log\pi_\phi(a_t|s_t)])^2]. \tag{25}$$

As mentioned before, the policy network needs to obtain the mean and standard deviation of the Gaussian distribution, but the corresponding mean and standard deviation are not differentiable. As such, we use DNN to reparameterize the policy:

$$a_t = f_\phi(\epsilon_t; s_t), \tag{26}$$

where ϵ_t is a random variable sampled from a fixed prior distribution, such as a spherical Gaussian distribution. In this setting, instead of sampling directly from the mean and standard deviation, the network is first sampled from a Gaussian distribution and then multiplied by the standard deviation plus the mean to make the network differentiable. As a result, the loss function can be rewritten as:

$$\nabla_\phi J(\pi_\phi) = \mathbb{E}_{s_t\sim\mathcal{R},\epsilon_t\sim\mathcal{N}}[\min_{j=1,2} Q_{\theta_j}(s_t, f_\phi(\epsilon_t;s_t)) - \log\pi_\phi(f_\phi(\epsilon_t;s_t)|s_t)], \tag{27}$$

where π_ϕ is defined implicitly in terms of f_ϕ, and we have noted that the partition function is independent of ϕ and can thus be omitted.

Table 1
Comparison of different single-agent reinforcement learning algorithms.

Method	On-/Off-policy	Value-/Policy-based	Policy type	State space	Action space
Q-learning	Off	Value	Deterministic	Discrete	Discrete
DQN	Off	Value	Deterministic	Continuous	Discrete
PG	On	Policy	Stochastic	Continuous	Discrete and/or Continuous
PPO	On	Policy	Stochastic	Continuous	Discrete and/or Continuous
DDPG	Off	Policy + Value	Deterministic	Continuous	Continuous
SAC	Off	Policy + Value	Stochastic	Continuous	Discrete and/or Continuous

2.2.7. Comparison of single-agent reinforcement learning algorithms

Having described the principles and the mathematical equations of different RL algorithms in Sections 2.2.1–2.2.6, the objective of this subsection is firstly to make a comparison between the six described RL algorithms from different perspectives in Table 1; and secondly, to investigate the network structures of six RL algorithms in Fig. 4.

It can be found from Table 1 that Q-learning, DQN, DDPG, and SAC are all categorized into the off-policy algorithm, which means they are allowed to update the current policy using the transitions from old policies, i.e., sampling transitions (or experiences) from the replay buffer to calculate policy updates and can be reutilized. As a result, the sampled transitions are mixed that are generated by different policies, which can improve the sample efficiency. However, off-policy algorithms are not motivated by policy improvement guarantees and do not directly control the bias introduced by off-policy data. Meanwhile, PG and PPO belong to the on-policy category, which means they are allowed to update the policy based on the transitions generated by the current policy. The critic network can make a more accurate value prediction for the current policy network. However, the on-policy algorithms suffer from poor sampling efficiency, since the prior transitions cannot be utilized frequently to update the policy network. To this end, importance sampling is normally deployed to improve the sample efficiency and stability of on-policy algorithms.

In general, there are three approaches to representing and training agents with RL. The first one is the value-based RL algorithm that learns the Q-value for the optimal action-value function $Q^*(s,a)$, such as Q-learning and DQN. Typically, they use an objective function based on the Bellman optimality equation. The second one is the policy-based RL algorithm that learns the policy directly, such as PG and PPO. They use gradient ascent to optimize the parameters ϕ directly on the performance objective $J(\pi_\phi)$. Policy-based algorithms also usually involves learning an approximator $V_\psi(s)$ or $Q_\theta(s,a)$ for the on-policy value function $V^\pi(s)$ or $Q^\pi(s,a)$, which is then used to determine how to update the policy π_ϕ. Here, we try to discuss the trade-offs between value-based and policy-based algorithms. The primary strength of policy-based algorithms is that they are principled in the sense that agents can directly optimize for the thing they want. This tends to make them stable and reliable. By contrast, value-based algorithms only indirectly optimize for agents' performance by training $Q(s,a)$ to satisfy a self-consistency equation. There are many failure modes for this kind of learning, so it tends to be less stable. However, value-based algorithms gain the advantage of being substantially more sample efficient when they do work, because they can reuse data more effectively than policy-based algorithms. In this context, the third one is the algorithm that interpolates between value-based and policy-based algorithms, such as DDPG and SAC.

In RL algorithms, there are the concepts of stochastic and deterministic policies. A deterministic policy is a function of the form $\pi(s) : \mathcal{S} \to \mathcal{A}$, i.e., a function from the set of environment states $\mathcal{S}$, to the set of action $\mathcal{A}$. For example, the set of actions is composed of the actions $\mathcal{A} = \{\text{charging}, \text{discharging}\}$. Unless the policy changes, $\pi(s)$ is always the same action (e.g., "charging") given a state, $s \in \mathcal{S}$. In Table 1, Q-learning, DQN, and DDPG all belong to the category of deterministic policy RL algorithm. A stochastic policy can be described as a family of conditional probability distributions from the set of states to the set of actions $\pi(a|s) : \mathcal{S} \times \mathcal{A} \to [0,1]$. A probability distribution is a function that assigns a probability for each action ("charging" and "discharging") and such that the sum of both the probabilities is 1.

Q-learning can operate only in discrete state and action spaces because it is based on Bellman back-ups and the discrete-space version of Bellman's equation. However, most EV applications of RL require a continuous state space defined by means of continuous variables such as battery status, electricity prices, etc. The usual approach has been to discretize the continuous variables, which quickly leads to combinatorial explosion and the well-known "curse of dimensionality". To this end, DQN, PG, PPO, DDPG, and SAC can deal with continuous state space via a DNN-based universal function approximator that does not use a discretization grid of the entire space. However, the value-based DQN still suffers from the discrete action space. As a result, PG, PPO, DDPG, and SAC, owing to their policy gradient theorem, can handle the continuous action space. Specifically, PG, PPO, and SAC are capable of constructing a Gaussian policy by learning the mean (sigmoid activate function) and standard division (softplus activate function), separately. Then, the continuous action can be directly sampled from the learned Gaussian policy, as depicted in Fig. 4(d). DDPG, owing to its deterministic policy gradient theorem, can directly compute the continuous action value instead of sampling from a distribution. Furthermore, the stochastic-based PG, PPO, and SAC can also generate the discrete actions by deploying the softmax activate function to the output layer, as depicted in Fig. 4(c).

Then, we investigate the structures of different RL algorithms in Fig. 4. First of all, Q-learning is characterized by a tabular learning approach; thus, there is no need for a DNN to approximate the Q-value. DQN can employ DNNs as a function approximator (parameterized by θ) that takes the state as input and the Q-values of all possible actions as outputs via the linear activation function. To model the stochastic policy, the softmax activation function is used to generate the selection probabilities of all possible actions via the stochastic policy gradient theorem, as shown in Fig. 4(c). In addition, the sigmoid and softplus activation functions can also be used to learn the mean and standard deviation of a Gaussian policy, as shown in Fig. 4(d). In this setting, the continuous action value can be sampled from the learned Gaussian policy. Unlike policy-based networks, critic network predicts the value of the importance of being in a state (state-value) or for an action–state pair (Q-value). As a result, PPO and DDPG are introduced via the critic network to learn the state-value function $V_\psi(s)$ in Fig. 4(e) and the Q-value function $Q_\theta(s,a)$ in Fig. 4(f), respectively. Furthermore, PPO uses the softmax activation function to learn a stochastic policy $\pi_\phi(a|s)$, whereas DDPG uses the sigmoid activation function to learn a deterministic policy $a = \mu_\phi(s)$. Finally, SAC learns a policy (actor) network $\pi_\phi(s)$, two Q-value networks $Q_{\theta_1}(s,a), Q_{\theta_2}(s,a)$, and a state-value network $V_\psi(s)$ in Fig. 4(g). Since SAC is an off-policy algorithm, a replay buffer is used to store the past experiences and update the networks for more advanced sampling efficiency. To estimate the policy evaluation, SAC, unlike PPO and DDPG, learns both the state-value function $V_\psi(s)$ and the Q-value function $Q_{\theta_{j=1,2}}(s,a)$. Furthermore, SAC learns two Q-value functions $Q_{\theta_1}(s,a)$ and $Q_{\theta_2}(s,a)$ and then takes the smaller of the two for training state-value function $V_\psi(s)$ and policy $\pi_\phi(s)$. This technique can eliminate the overestimation of Q-value so as to stabilize the training performance.

Finally, it should be mentioned that PPO and SAC can model both the discrete and continuous action spaces by employing different activation functions, e.g., softmax for discrete action space in Fig. 4(c); sigmoid and softplus for continuous action spaces in Fig. 4(d).

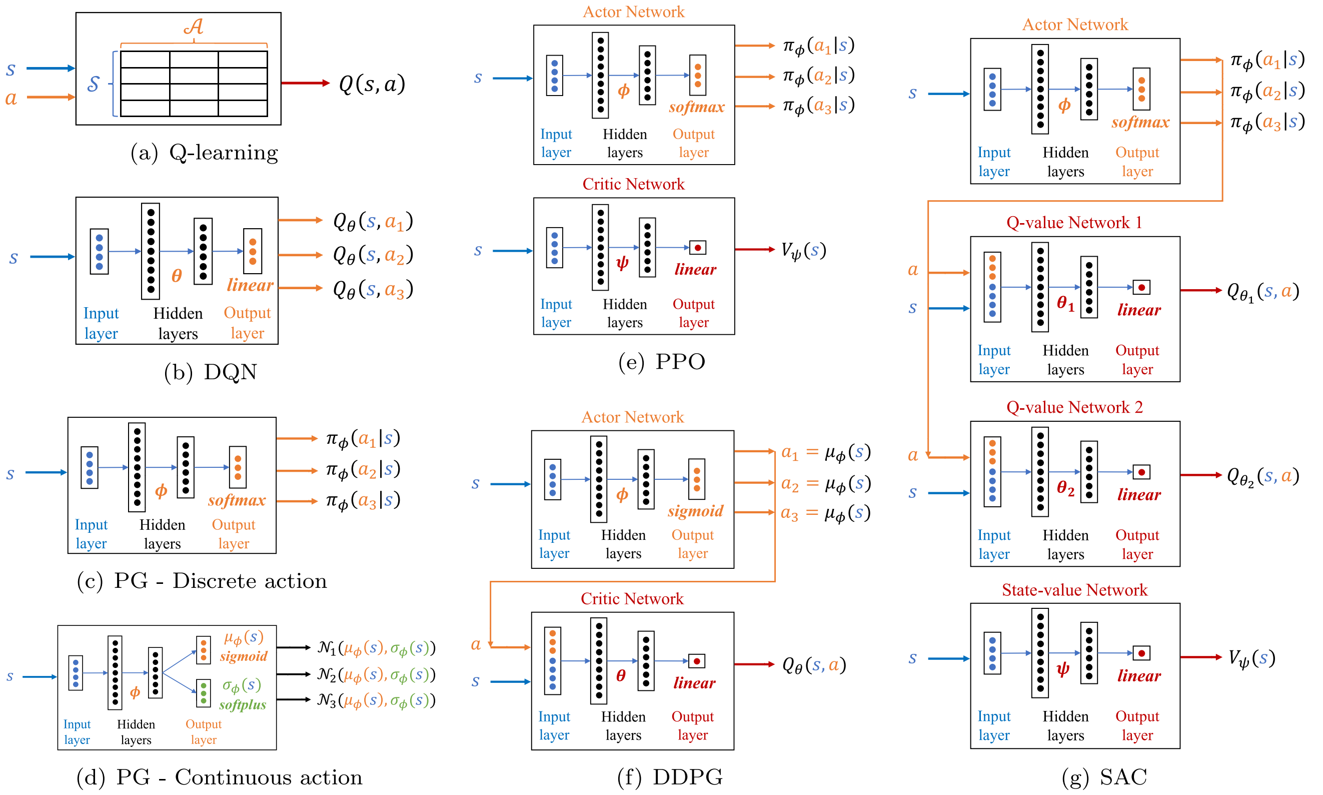


Fig. 4. Network structures of (a) Q-learning, (b) deep Q-network (DQN), (c) policy gradient (PG) with discrete action, (d) policy gradient (PG) with continuous action, (e) proximal policy optimization (PPO), (f) deep deterministic policy gradient (DDPG), and (g) soft actor–critic (SAC).

Table 2
Network structures and models for three multi-agent reinforcement learning algorithms.

Method	Actor network	Critic network	Number of networks	Training	Execution
CTCE	$\pi_\phi(a_{1:\mathcal{I}}\|o_{1:\mathcal{I}})$	$Q_\theta(o_{1:\mathcal{I}},a_{1:\mathcal{I}})$	2	Centralized	Centralized
DTDE	$\pi_{\phi_i}(a_i\|o_i)$	$Q_{\theta_i}(o_i,a_i)$	$\|\mathcal{I}\|\times 2$	Decentralized	Decentralized
CTDE	$\pi_{\phi_i}(a_i\|o_i)$	$Q_{\theta_i}(o_{1:\mathcal{I}},a_{1:\mathcal{I}})$	$\|\mathcal{I}\|\times 2$	Centralized	Decentralized

2.3. Multi-agent reinforcement learning

In the context of RL, complex applications require the involvement of multiple agents to learn and process different tasks simultaneously, thus rendering SARL into multi-agent reinforcement learning (MARL). In any MARL algorithm, the MDP is generalized to a Markov game, which can be defined by a tuple $\langle\mathcal{I},\mathcal{S},\mathcal{O}_i,\mathcal{A}_i,\mathcal{R}_i,\mathcal{T},\gamma\rangle$. Specifically, a Markov game representing a group of agents $i\in\mathcal{I}$ interacting with the environment (Fig. 5) that includes a collection of global states $s\in\mathcal{S}$, a collection of local observations $\{o_i\in\mathcal{O}_i\}$, a collection of action sets $\{a_i\in\mathcal{A}_i\}$, and a collection of reward functions $\{r_i\in\mathcal{R}_i\}$, as well as a state transition function $\mathcal{T}(s,a_\mathcal{I})$. For each agent i at time step t, an action $a_{i,t}$ is computed using the policy $\pi_i(a|o)$ conditioned on the current local observation $o_{i,t}$. Then, the environment transits to the next state given the transition function $\mathcal{T}(s_{t+1}|s_t,a_{\mathcal{I},t})$, while agent i receives a rewarded $r_{i,t}$ and the next local observation $o_{i,t+1}$. Following this process, each agent i receives a trajectory of local observations, actions, and rewards: $\tau_i=o_{i,1},a_{i,1},r_{i,1},o_{i,2},\dots,r_{i,T}$ over $\mathcal{O}_i\times\mathcal{A}_i\times\mathcal{R}_i\rightarrow\mathbb{R}$. In the simplest case, s could consist of the local observations of all agents, $s=(o_1,\dots,o_I)$, however it could also include additional state information if available [32]. The objective of each agent i is maximizing its cumulative discounted reward $R_i=\sum_{t=0}^{T}\gamma^t r_{i,t}$, where $\gamma\in[0,1)$ and $T=24$ h are the discount factor and daily horizon, respectively.

However, different from the MDP in SARL algorithms, the agents in the Markov game are coupled with each other in the environment and can influence the environment dynamics and the optimal policies. In other words, it is more challenging to learn the optimal policies for all agents and manage the interactions between them, since the agents' policies are implicitly formulated as part of the environment dynamics while their policies are continuously adjusted during the training process, thereby easily suffering from instability issues. This section aims at introducing three typical frameworks that are widely used in MARL algorithms: (1) centralized training with centralized execution (CTCE); (2) decentralized training with decentralized execution (DTDE); and (3) centralized training with decentralized execution (CTDE). Fig. 6 illustrates the workflows of these three frameworks based on the conventional actor–critic architecture of DDPG algorithm [25], in which their network structures and detailed information are compared in Table 2.

2.3.1. Centralized training with centralized execution (CTCE)

It can be observed from Fig. 6(a) that the CTCE framework is managed by a central controller, which deploys a Q-value (critic) network $Q_\theta(o_{1:\mathcal{I}},a_{1:\mathcal{I}})$ parameterized by θ and a policy (actor) network $\mu_\phi(o_{1:\mathcal{I}})$ parameterized by ϕ. As a result, the training and execution processes are both done by the central controller that requires the information of all agents' local observations $o_{1:\mathcal{I}}$ and helps all agents make actions $a_{1:\mathcal{I}}$. Each agent i in CTCE is responsible for interacting with the environment, executing the action a_i generated by the central controller, and reporting the local observation o_i and the reward r_i to the central controller. As a result, at time step t, the central controller can collect all of the agents' local observations as the state $s_t=[o_{1,t},o_{2,t},\dots,o_{I,t}]$, as well as the sum of all of the agents' local rewards

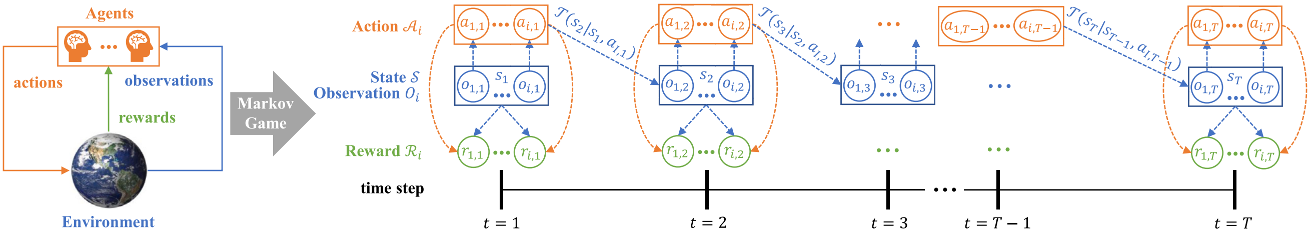
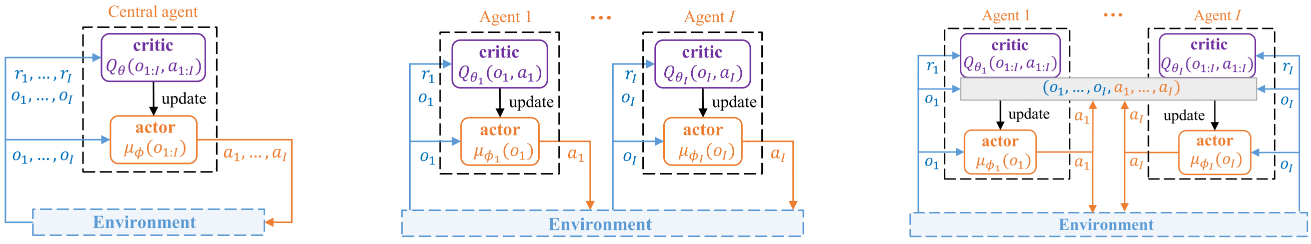


Fig. 5. Agents–environment interactions and the process of Markov Game.



(a) CTCE (b) DTDE (c) CTDE

Fig. 6. Workflows of three multi-agent reinforcement learning frameworks: (a) centralized training with centralized execution (CTCE), (b) decentralized training with decentralized execution (DTDE), (c) centralized training with decentralized execution (CTDE).

as the central controller's reward $r_t = r_{1,t}+r_{2,t}+\cdots+r_{I,t}$. To that end, the critic network's parameters θ are optimized to minimize the TD error, which is defined as:

$$\mathcal{L}(\theta) = \mathbb{E}_{o_{1:I,t}\sim\rho^\mu, a_{1:I,t}\sim\pi_\phi}[(Q_\theta(o_{1:I,t},a_{1:I,t}) - y_t)^2], \tag{28}$$

where the target value is:

$$y_t = \sum_{i=1}^{I} r_{i,t} + \gamma Q_{\theta'}(s_{1:I,t+1}, \mu_{\phi'}(s_{1:I,t+1})). \tag{29}$$

The parameters ϕ of the actor network are optimized using the deterministic policy gradient theorem with the computed gradient defined as:

$$\nabla_\phi J(\mu_\phi) = \mathbb{E}_{o_{1:I,t}\sim\rho}[\nabla_\phi\mu_\phi(o_{1:I,t})\nabla_{a_{1:I,t}}Q_\theta(o_{1:I,t},\mu_\phi(o_{1:I,t}))]. \tag{30}$$

At each time step t, the central controller collects the all agent's local observations $o_{1:I,t}$ and then makes actions $a_{1:I,t}$ using the policy network deployed by the central controller:

$$a_{1:I,t} = \mu_\phi(\cdot|o_{1:I,t}). \tag{31}$$

In this setting, the agents only need to execute the actions generated by the central controller and do not need to learn the policies by themselves. The reason for this is that the policy function $\mu_\phi(\cdot|o_{1:I,t})$ requires the global state of all agents' local observations as input, whereas individual agents do not know the global state and are not capable of taking actions on their own.

The advantage of the CTCE framework is that it is easy to implement as the conventional DDPG algorithm. Furthermore, its correctness can be guaranteed due to the centralized training that makes use of the complete state and action information as well as the centralized execution in observing the global state. However, CTCE suffers from a latency issue, which may affect the speed of both training and execution processes. In the framework of centralized execution, agent i transmits its local observation $o_{i,t}$ to the central controller, who generates actions $a_{1:I,t}$ only after collecting all local observations $o_{1:I,t}$. Then, the action $a_{i,t}$ also has to be transmitted to agent i. This process is usually slow, making real-time decision-making problems impractical. More importantly, the CTCE framework is centralized in both training and execution, and leads to an exponential growth in both local observation and action spaces with the number of agents, which raises the curse of dimensionality [22] and quickly becomes intractable for a large-scale multi-agent setup. Furthermore, the implementation of the CTCE framework may raise agents' opposition, since they are generally unwilling to reveal their private information and exchange such information with others.

2.3.2. Decentralized training with decentralized execution (DTDE)

The fundamental idea of the DTDE framework is to replace the global state s_t with the local observation $o_{i,t}$, approximating a policy (actor) network $\mu_{\phi_i}(o_i)$ and a Q-value (critic) network $Q_{\theta_i}(o_i,a_i)$ for each agent i. In this setting, the agents do not share parameters, i.e., $\phi_i \neq \phi_j, \theta_i \neq \theta_j, \forall j \in I \setminus \{i\}$. Both training and execution can be done locally by the agent without involving a central controller or any communication. To this end, the parameters θ_i of each agent's critic network can be optimized to minimize the TD error defined as:

$$\mathcal{L}(\theta_i) = \mathbb{E}_{o_{i,t}\sim\rho^\mu, a_{i,t}\sim\mu_{\phi_i}}[(Q_{\theta_i}(o_{i,t},a_{i,t}) - y_{i,t})^2], \forall i \in \mathcal{I}, \tag{32}$$

where the target value is:

$$y_{i,t} = r_{i,t} + \gamma Q_{\theta_i'}(o_{i,t+1}, \mu_{\phi_i'}(o_{i,t+1})), \forall i \in \mathcal{I}. \tag{33}$$

The parameters ϕ_i of the actor network of each agent i are optimized using the deterministic policy gradient theorem with the computed gradient defined as:

$$\nabla_{\phi_i} J(\mu_{\phi_i}) = \mathbb{E}_{o_{i,t}\sim\rho}[\nabla_{\phi_i}\mu_{\phi_i}(o_{i,t})\nabla_{a_{i,t}}Q_{\theta_i}(o_{i,t},\mu_{\phi_i}(o_{i,t}))], \forall i \in \mathcal{I}. \tag{34}$$

After completing the training, agent i no longer requires its critic network $Q_{\theta_i}(o_i,a_i)$. The agent only needs to use its local policy network $\mu_{\phi_i}(o_i)$ to make action a_i without communication. Thus, decentralized execution is fast and can make real-time decisions. However, the agents in the Markov game influence each other, and the decentralized training in the DTDE framework regards the agents as independent entities while ignoring the correlation between them and directly training each agent independently with the SARL algorithm. As a result, using the DTDE framework to solve the MARL problem is often ineffective in practice.

2.3.3. Centralized training with decentralized execution (CTDE)

The previous two frameworks discuss fully centralized and fully decentralized methods, and both implementations have their own advantages and disadvantages. Recently, the more popular MARL framework is CTDE, in which a central controller is used to assist the agent in

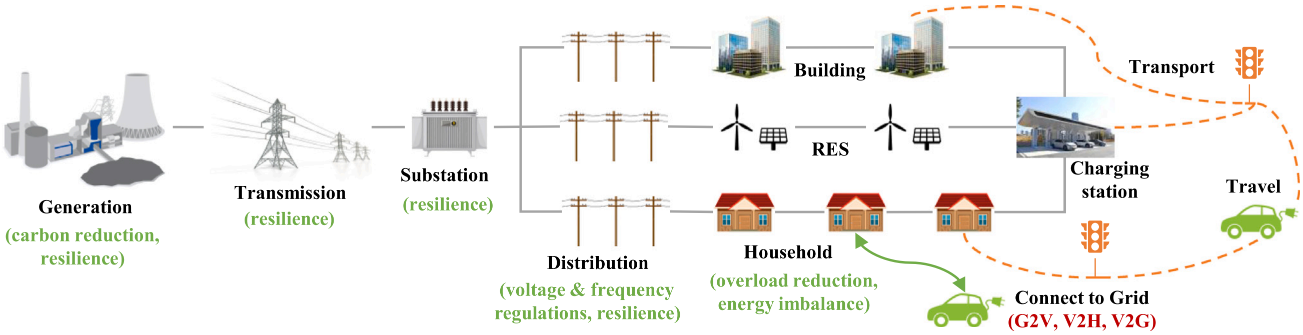


Fig. 7. The applications of EVs in power systems.

training; after training, the central controller is no longer needed, and each agent makes actions independently based on local observation o_i. Similar as DTDE, each agent in CTDE has a policy (actor) network $\mu_{\phi_i}(a_i|o_i)$ that inputs the local observation o_i and outputs the executed action a_i. However, the value (critic) network $Q_{\theta_i}(o_{1:I}, a_{1:I})$ adopts the centralized training of CTCE that requires the information of all agents' local observations $o_{1:I}$ and actions $a_{1:I}$. To this end, the parameters θ_i of each agent's critic network can be optimized to minimize the TD error defined as:

$$\mathcal{L}(\theta_i) = \mathbb{E}_{o_{1:I,t}\sim\rho^{\mu}, a_{i,t}\sim\mu_{\phi_i}}[(Q_{\theta_i}(o_{1:I,t}, a_{1:I,t}) - y_{i,t})^2], \forall i \in \mathcal{I}, \tag{35}$$

where the target value is:

$$y_{i,t} = r_{i,t} + \gamma Q_{\theta_i'}(o_{1:I,t+1}, \mu_{\phi_1'}(o_{1,t+1}), \dots, \mu_{\phi_I'}(o_{I,t+1})), \forall i \in \mathcal{I}. \tag{36}$$

The parameters ϕ_i of the actor network of each agent i are optimized using the deterministic policy gradient theorem with the computed gradient defined as:

$$\nabla_{\phi_i} J(\mu_{\phi_i}) = \mathbb{E}_{o_{i,t}\sim\rho}[\nabla_{\phi_i}\mu_{\phi_i}(o_{i,t})\nabla_{a_{i,t}} Q_{\theta_i}(o_{1:I,t}, \mu_{\phi_1}(o_{1,t}), \dots, \mu_{\phi_I}(o_{I,t}))], \forall i \in \mathcal{I}. \tag{37}$$

This framework effectively circumvents the challenge of environmental non-stationary during the training process when knowing the information of all agents' local observations and actions. During test time, the critic network is not needed and the policy execution is fully decentralized through each agent's actor network that only takes as input its own local observation. Nevertheless, as in centralized training, CTDE is not privacy-preserving and also suffers from a similar curse of dimensionality to CTCE, which is problematic in practical large-scale multi-agent applications. If the considered Markov game consists of agents with the same observation, action, and reward function, their policies can be trained with enhanced efficiency by using a parameter-sharing (PS) technique [33]. Specifically, PS allows all agents to share the parameters of a single control policy. This enables the shared policy to be trained with the sample experiences gathered by all agents, while still allowing different behaviors among different agents since each agent receives different local observations.

3. Applications of RL on EV dispatch problems

A typical EV dispatch problem mainly focuses on how to charge power efficiently and economically when EVs are connected to charging stations, such that EVs can have sufficient energy for their daily journeys, e.g., traveling from home to the office in the morning and back home from the office in the evening. In addition to the charging requirement, EVs can also discharge power to homes (V2H) or the electricity grid (V2G) for various ancillary service provisions, e.g., overload reduction, energy imbalance service, carbon intensity service, voltage and frequency regulations, etc. For instance, as shown in Fig. 7, EVs can choose to charge power from the home charging stations during the night for the coming day's traveling energy requirement or discharge power to the grid for certain ancillary service provisions. When V2H or V2G technologies are activated, it is more possible for EV users to make energy arbitrage through smart charging and discharging behaviors.

As discussed in Section 1, RL demonstrates the benefits of dealing with real-time stochastic and dynamic problems without the need for knowledge of system models and technical parameters. So far, RL algorithms have been successfully applied to various EV dispatch problems, including EV charging towards cost minimization, EV charging/discharging towards overload reduction, EV ancillary service provisions (e.g., energy balance, frequency, voltage, etc.), as better summarized in Tables 3, 4, 5, and 6, respectively. To apply RL algorithms to the various EV dispatch problems, the common practice in the existing work is to first formulate them as an MDP solved by SARL algorithms or a Markov game solved by MARL algorithms. The following subsections then provide the detailed components of the MDP or Markov game corresponding to different EV applications.

3.1. RL for G2V

Transport sectors are regarded as one of the largest contributors to the excessive use of oil resources and various environmental problems (e.g., pollution and climate change) [34]. In order to address these issues, many countries have passed regulations to restrict the fossil fuel consumption of traditional vehicles, promising a low-carbon future, which has led to a rapid increase in the use of EVs in the transport sector [35]. As an alternative to traditional vehicles, travel requirements should be firstly satisfied for EV owners through effective EV charging behaviors when they are connected to the grid through local charging stations [36].

Note that uncontrolled EV charging behaviors can influence the normal travel of users and lead to a significant increase in operating costs. From the electricity market perspective, many utilities have started offering time-varying electricity price signals, which allows EV owners to select appropriate timeslots for more economical charging [37]. In other words, EVs can make use of electricity price variations and demand flexibility to shift the charging power to the periods when electricity prices and/or grid demand are relatively low. For this reason, much research has focused on developing RL-based smart charging schemes for EVs towards charging cost minimization but also ensuring travel requirements, as summarized in Table 3. The most commonly-used state/observation, action, and reward function have been detailed as follows:

3.1.1. State

Since the EV charging problems involve both transport and power sectors, the local observations $o_{i,t}$ of EV agent i at time step t shall include the information from both sectors, which can be generalized as:

$$o_{i,t} = [t, \Delta_{i,t}, L^{rd}_{i,t}, V^{rd}_{i,t}, S_{i,t}, \lambda^{g}_{t}, G^{res}_{i,t}, D^{p}_{i,t}, D^{ev}_{i,t}], \forall i \in \mathcal{I}, \forall t \in T, \tag{38}$$

Table 3
Summary of G2V.

Ref.	State	Action	Reward	Algorithm	Key features
[38]	Time, RES, load, weather, traffic, waiting time, etc.	Charging rate	Minimize charging cost, non-completion penalty	SAC, PPO, DDPG	Combine the advantages of optimal control policy characterizations and model-free DRL algorithm.
[39]	Travel energy requirement, charging time, price signals	Charging and discharging rates	Minimize charging cost, non-completion penalty	DDPG, DQN	Use two replay buffers to address the limitations of sparse rewards, use LSTM to extract time-series price information.
[37]	Time, aggregate charging demand	Charging rate	Minimize charging cost, non-completion penalty	Fitted Q-iteration	Contribute a new MDP formulation with a scalable state representation independent of the number of charging stations.
[40]	Time, SoC, electricity price	Daily charging energy	Minimize charging cost	Fitted Q-Iteration	A Bayesian neural network is employed to predict the electricity prices; a linear program is used to optimally schedule the PEV battery charging.
[41]	Time, SoC, RES, charging rate, charging time, electricity price	Charging rate	Minimize charging cost and wind power fluctuation	DDPG	Realize the optimized EV charging control under uncertain wind power, electricity price, charging pile, and user requirements.
[42]	Time, SoC, RES, energy requirement, charging demand, charging time, electricity price	Startup time of charging pile	Improve charging satisfaction, reduce operation cost and PV curtailment	DQN	Propose a charging station scheduling strategy that combines EV random charging behavior characteristics with DRL algorithm.
[43]	Residual charging demand and parking time	Charging price and charging rate	Maximize charging station profit	SARSA	Develop a model-free data-driven method for joint pricing and charging scheduling at an EV charging station with random EV arrivals and departures.
[44]	Queuing system capacities, arrival rate, completed time periods	Charging price	Improve the quality of service	Actor-critic Q-learning	Propose a new dynamic pricing framework for EV charging stations that can offer multiple charging options to customers over a finite time horizon.
[45]	PV generation, building demand, SoC, departure time	Charging rate	Maximize PV self-consumption, achieve the highest SOC	DDQN, P-DQN, DDPG	Propose three mathematical formulations of the problem in the form of MDPs that differ by the type of action space.
[46]	Travel energy requirement	Charging rate	Minimize charging cost	Fitted Q-iteration	EV charging is controlled by a heuristic scheme, and the resulting charging behavior is learned by batch RL.
[47]	RES, DG output, SoC, temperature, charging demand, AC status	Charging rate	Minimize DG generation, EV charging, and battery degradation cost, ensure user comfort	DDPG, A2C	Propose a two-layer interactive architecture for effective control while preserving the user privacy data.
[48]	Wind power output and traffic demand	Charging service fee	Minimize total social cost	DDPG, Fitted Q-iteration	Develop a RL framework to decouple and solve the stochastic EV charging problem in a coupled power-transport network.
[49]	EV location, SoC, traffic condition, electricity price, waiting time	Designated charging station, planned route	Minimize battery consumption, travel time, charging cost and waiting time.	DQN	State features are extracted out of collected data including traffic condition, charging price and waiting time via a shortest charging route model.
[50]	EV arriving time, number of charging piles	Scheduled EV, chosen station, charging mode	Minimize charging time and travel distance	DQN	Aim to reduce total charging time and travel distance of EVs charging en-route.
[51]	Time, charging station number, location, road length, SoC	Charging station choice	Minimize charging cost and travel time	Rainbow DQN	Propose a new platform for real-time EV charging navigation based on graph RL.
[52]	Time, location, SoC, charging station capacity, the number of charging and waiting EVs, road speed, nodal voltage	Charging station choice	Minimize estimated time cost	DQN	Propose a physical connection-based graph formulation method with feature projection to integrate multi-dimensional information into a graph.
[53]	Storage capacity, charging demand, electricity price	Charging and discharging rate	Minimize charging cost, over charging loss and increase pre-charge benefits	CommmNet, DQN, PPO	Use MARL algorithm for energy management of charging stations in a distributed manner under dynamic time-varying PV generation.
[54]	Time, SoC, location, charging interval	Pass, charge, assign	Minimize charging cost and travel cost	MADQN	Propose a novel framework with decentralized learning and centralized decision making for EV ride-hailing.

(*continued on next page*)

Table 3 (*continued*).

Ref.	State	Action	Reward	Algorithm	Key features
[55]	Time, SoC, location	Rebalancing or charging decision	Charging cost	MADQN	Consider joint charging scheduling, order dispatching, and vehicle rebalancing for large-scale shared EV fleet operator.
[56]	Time, SoC, number of charging piles, charging demand	Bidding quantity and price	Minimize charging cost	MADQN	Propose a multi-agent DQN algorithm to learn the optimal bidding strategy for multiple EVs in an auction market.
[57]	Time interval, charging load, historical charging price	Charging price	Maximize charging profit	MASAC	Propose a strategic charging pricing scheme for charging station operators based on a non-cooperative Stackelberg equilibrium framework.
[58]	Each player's rationality and charging price	Charging price	Maximize charging profit	MADQN	Use MARL to model the pricing competition of multiple charging stations in transport networks with elastic traffic demands.

where $o_{i,t}$ consists of three parts: (1) the time information of current time step t and charging interval $\Delta_{i,t}$ of EV agent i at time step t; (2) the transport information of EV location $L^{rd}_{i,t}$ and traffic volume $V^{rd}_{i,t}$; and (3) the power information of battery state-of-the-charge (SoC) level $S_{i,t}$, grid electricity price λ^{g}_{t}, nodal renewable generation $G^{res}_{i,t}$, nodal power demand $D^{p}_{i,t}$, and EV charging demand $D^{ev}_{i,t}$. Among them, t, λ^{g}_{t}, $G^{res}_{i,t}$, and $D^{p}_{i,t}$ belong to exogenous states that represent the local information unaffected by actions, while $\Delta_{i,t}$, $L^{rd}_{i,t}$, $V^{rd}_{i,t}$, $S_{i,t}$, and $D^{ev}_{i,t}$ correspond to endogenous states that serve as the feedback signals of executed routing and scheduling actions by EV agent i.

3.1.2. Action

Similarly, the action $a_{i,t}$ of EV agent i at time step t also involves two parts corresponding to both transport and power sectors, which can be generalized as:

$$a_{i,t} = [a^{tsp}_{i,t}, a^{pow}_{i,t}], \forall i \in \mathcal{I},\ \forall t \in T, \tag{39}$$

where (1) the discrete routing action $a^{tsp}_{i,t} \in \{0, 1, \ldots, N^{rd}\}$ is selected from the set of potential routes upon the transport node, in which 0 denotes no routing behaviors and N^{rd} denotes the number of available commuting routes at current transport node; and (2) the continuous charging action $a^{pow}_{i,t} \in [0, 1]$ represents the magnitude of charging rate of EV agent i as a percentage of its power capacity $\overline{P}_{i}$. It is worth noting that discharging behaviors are not considered in the G2V problem.

3.1.3. Reward function

At the end of time step t, the EV agent i obtains its reward $r^{c}_{i,t}$. The objective of EV agent i is to minimize charging costs while also ensuring a sufficient energy requirement for travel. As such, the reward function $r_{i,t}$ in Eq. (40) can be designed as two parts conditioned on: (1) the cost of charging power $P_{i,t}$ when EV agent i is connected to the grid ($A_{i,t} = 1$) in the power network at time step t; (2) the penalty of insufficient charging upon departure in the transport network, i.e., $E_{i,t} = S_{i,t}\overline{E}_{i} \geq E^{tp}_{i}$ may not be satisfied when EV is traveling ($A_{i,t} = 0$).

$$r^{g2v}_{i,t} = \begin{cases} -\lambda^{g}_{t} P^{c}_{i,t} & \text{if } A_{i,t} = 1 \\ \kappa [E_{i,t} - E^{tp}_{i,t}]^{-} & \text{if } A_{i,t} = 0 \end{cases}, \forall i \in \mathcal{I},\ \forall t \in T, \tag{40}$$

where κ is a penalty factor to penalize the extent of constraint violation.

3.1.4. Discussion

Primary research limitations and potential solutions to the RL-based G2V problems have been listed and discussed as follows:

3.1.4.1. Detailed transportation network models.

- **Research limitations**: As described in Table 3, most existing work uses certain probability distributions or real-world datasets to capture random EV arrival and departure behaviors, SoC levels, and required charging demand for uninterruptible daily journeys. However, these papers do not consider the model details of transport networks and real-time EV routing behaviors, which can be unrealistic and lead to inaccurate results. In fact, only a few references (e.g., [48–50]) apply RL-based methods for both EV routing and scheduling characteristics. It is worth noting that these routing and scheduling decisions are mutually influenced, since an efficient routing decision can better avoid traffic congestion and save more time for EVs to exploit their energy flexibility for ancillary service provision, while an efficient power scheduling decision can ensure a high battery SoC level for EVs to exploit their transportation mobility. As such, to obtain accurate charging intervals and charging demand, it is necessary to capture both the routing and scheduling characteristics of EV fleets in RL setups.
- **Potential solutions**: To effectively apply RL algorithms to both the routing and scheduling behaviors of EVs, the main challenge is related to the hybrid discrete and continuous action domains, while classical RL algorithms are mainly applied in either discrete or continuous action domains. Therefore, it is difficult to use one RL policy to capture both the routing and scheduling actions of EVs. To address this challenge, there has been research employing the hybrid RL algorithm (e.g., [8]) and hierarchical RL algorithm (e.g., [81]) for EV dispatching problems, which are capable of capturing both discrete and continuous actions. However, the scalability and reliability of hybrid RL algorithms should be further investigated, especially when a multi-agent setup is required for large-scale EV fleets.

3.1.4.2. Effective reward function designs.

- **Research limitations**: The reward function for EV dispatch problems involves different perspectives. In more detail, charging cost and the penalty for non-completion are two basic elements to ensure successful EV daily journeys. Note that the model-free RL algorithm cannot handle the traveling energy requirement constraint in a mathematical manner like the model-based optimization approach [74]. In general, introducing appropriate weighting factors is necessary to fuse several different objectives into one reward function and differentiate their prioritization. However, determining the values of weighting factors for a trade-off can be very difficult due to the lack of accurate knowledge about the priority of different objectives. As such, a sensitivity analysis may be required to further evaluate the impact of penalty factors on the trained policy and identify the suitable selection for their values in EV dispatch problems, which is not a trivial task.
- **Potential solutions**: To better capture different objectives in the reward function design process, two potential solutions are available in existing work: (i) applying multi-objective RL methods (e.g., pareto reinforcement learning) [89] on EV dispatch problems, which can output a pareto front rather than a single solution and then effectively eliminate the need for weighting

Table 4
Summary of V2H.

Ref.	State	Action	Reward	Algorithm	Key features
[59]	Time, SoC, load, EV number, charging requirement	Charging rate	Maximize charging reward and departure reward	DQN	Address a simple but scalable smart charging coordination strategy for EVs with forward-looking charging schedules.
[60]	Residual demand, remaining time, current time	Charging rate	Minimize the penalty over the aggregate charging rate	PPO	Extend the action space to be consistent and state-independent for network training, and revise the reward function to penalize the neural network output.
[61]	Time, occupancy of parking lot, EV cluster, charging rate, power threshold	Charge or not	Maximize transferred energy without violating power threshold	DDQN	Ensure the completion of charging transactions in a timely manner while reducing demand peaks.
[62]	Time, electricity price, SoC, charging demand	Charging and discharging rate	Maximize power scheduling profit, minimize non-completion penalty	DQN	Consider a public charger shared among multiple users, estimate probability density functions from EV charging data using kernel density estimation.
[63]	Electricity price, demand	Charging price	Maximize charging profit	DDPG	Establishes a quarter-hourly V2G dynamic time-sharing pricing model based on DDPG.
[64]	Time, SoC, electricity price	Charging and discharging rate	Minimize EV charging costs, peak-cutting, valley-filling, meet charging demand	DDPG	Propose a distributed real-time scheduling optimization structure and establish a scheduling model of a single EV agent.
[65]	Location, load, solar irradiance, SoC	Moving direction, charge/idle/ discharge	Minimize charging cost	DQN	Address uncertainties in power supply and demand by dispatching EVs to supply energy for consumers at different locations.
[66]	Time, SoC, location, transformer load	Charge or not	Minimize charging cost, increase user satisfaction and avoid overload	MASCO	Build a multi-objective architecture in a distributed manner, aiming at minimizing energy costs and avoiding overloads, while allowing EV recharging.
[7]	Time, location, SoC, charging demand, electricity price	Charging and discharging rate	Minimize charging cost and drivers' range anxiety, avoid overload	MASAC	Formulates the EVs charging problem as a Markov game with an unknown transition function and propose a cooperative charging control strategy.
[67]	Time, location, SoC	Charging, find passengers	Finding passengers and minimize charging cost	MAQ-learning	Define the charging loads of plug-in electric taxis in both the temporal and spatial scales.
[68]	Time, SoC, charging preference, temperature, electricity price	Charging or discharging rate	Minimize the transformer loss and the EV dissatisfaction	MATD3, TD3	A centralized evolutionary curriculum learning mechanism is adopted to enhance the coordination of multiple EVs.
[69]	Time, charging power, congestion signals	Charging and discharging rate	Maximize the substation loading	MASAC	Propose an adaptive control algorithm for plug-in EV charging without straining the power system.
[70]	Time, EV type, SoC, load, electricity price	Charging and discharging rate	Minimize charging cost, range anxiety, transformer loss, battery degradation	MASAC	Propose a decentralized EV charging framework for optimization of the loss of transformer life considering the dis-satisfactions of EV owners.

factors; (ii) formulating the RL-based EV dispatch problems as constrained policy optimization algorithms [90,91], which can handle the reward and the constraints independently and do not need to carefully design specific reward functions for constraint violations.

3.2. RL for V2H

The growth of EVs brings viable solutions for future low-carbon power systems. Nevertheless, the large-scale EV penetration into power systems can also significantly increase residential demand, leading to the potential overload of distribution grid transformers or even premature failures. Specifically, uncoordinated EV charging could raise the current peak demand level or cause a new peak demand by changing the profile of the system demand. As a result, the system operator must appropriately coordinate the charging and discharging behaviors of large-scale EV fleets to reduce the risk of overloading distribution networks via vehicle-to-home (V2H) technologies.

Coordinating the charging rates of EVs to flatten the load profile and reduce its variance is a non-trivial task because of the potential privacy issues and EV-related uncertainties (e.g., arrival and departure time, charging interval, SoC level, etc.). As a model-free approach, RL can encapsulate various uncertainties into the training procedure and assist EVs to reach a cooperative fashion within a decentralized framework for better privacy protection. There has been much research focused on using RL algorithms to avoid overloading issues caused by the integration of large-scale EVs, as better summarized in Table 4. The most commonly-used state, action, and reward function have been detailed as follows:

3.2.1. State

Given that the V2H problem only affects the power sector, the local observation $o_{i,t}$ of EV agent i at time step t can be generalized as follows:

$$o_{i,t} = [t, \Delta_{i,t}, S_{i,t}, \lambda_t^g, G_{i,t}^{res}, D_{i,t}^p, D_{i,t}^{ev}], \forall i \in \mathcal{I}, \forall t \in \mathcal{T}, \quad (41)$$

Table 5
Summary of V2G.

Ref.	State	Action	Reward	Algorithm	Key features
[71]	Time, SoC, utilization rate of charging piles, electricity price, station capacities	Charging or discharging rate	Maximize charging profit	DDPG, TD3, SAC	Construct a DRL based Stackelberg game model for a VPP with EV charging stations.
[72]	SoC, wind speed, station capacity, solar irradiance, trading volume, observed utility	Charging rate, bidding price	Minimize charging cost	DDPG	An asynchronous learning framework is put forward to help aggregators formulate bids, including bidding price and volume.
[73]	Time, SoC, electricity price, charging preference	Charging and discharging rate	Minimize charging cost and the driver's aggregate anxiety	SAC, DQN, TD3, PPO	Introduce an aggregate anxiety concept to characterize both the driver's anxiety on the EV's range and uncertain events.
[74]	Electricity price, flexible and inflexible EV demand	Retail prices	Maximize overall profit while avoid constraint violations	Q-learning, DQN, DDPG	Propose a DRL algorithm that sets up the problem in multi-dimensional continuous state and action spaces.
[75]	SoC, past 24-hour electricity prices	Charging and discharging rate	Maximize power scheduling profit	Constrained PG, DQN, DDPG	Propose a constrained charging and discharging scheduling strategy to minimize the charging cost as well as guarantee the EV can be fully charged.
[76]	EV location, SoC, electricity price	Charging and discharging rate	Minimize charging cost, degradation cost, range anxiety	DQN	Formulate the EV charging and discharging scheduling problem as an MDP from the user's perspective.
[77]	Time, SoC, electricity price, PV generation	Charging and discharging rate	Minimize charging cost and overcharging penalty, undercharging, and user preference	PG, A3C	Present a hierarchical DRL method for the scheduling of energy consumption of smart home appliances and DERs.
[78]	SoC, electricity price	Charging and discharging rate	Minimize charging cost and penalties of battery safety and travel requirement	DQN, DDPG	Combine the feature extraction ability of DL and the decision-making ability of RL for an EV charging strategy that reduces charging cost for the EV owner
[79]	Day, time, SoC, electricity price, EV status	Discharge, idle, or charge	Minimize charging cost, penalty of insufficient energy for travel	Q-learning, DQN	Propose a demand response method to reduce the long-term charging cost of single plug-in EV while overcoming obstacles from uncertainties.
[80]	EV aggregator suppliers' capacities	Bidding price	Maximize payoff after market clearing	MAQ -learning	Propose a competitive bidding strategy for wind power plants and EV aggregators in a pool-based day-ahead electricity market.
[81]	Location, traffic volume, demand, RES, SoC, electricity price, carbon intensity	Moving direction, charging or discharging rate, balance service provision	Maximize ancillary service profit, minimize travel time	HRL, MAPPO, PPO	Develop a MARL algorithm for cooperative EVs to optimize the provision of multiple interdependent services, including charging, demand management, carbon intensity, and balancing service.
[82]	Electricity price, SoC, charging demand, PV generation	Charging and discharging rate, and energy selling price schedule	Maximize power scheduling profit, minimize overcharging and undercharging penalties	FRL, SAC	Propose a privacy-preserving distributed RL framework that maximizes the profits of multiple smart charging stations integrated with photovoltaic and energy storage systems under a dynamic pricing strategy.
[8]	Electricity price, carbon intensity, SoC, RES, load, location, line status	Moving direction, charging and discharging rate	Maximize power scheduling profit, reduce load shedding	Hybrid MAPPO	Propose a MARL method to address the routing and scheduling problem of multiple EVs towards ancillary service provision and resilience control.

where $o_{i,t}$ consist of three parts: (1) the time information of current time step t and charging interval $\Delta_{i,t}$; and (2) the power information of battery SoC level $S_{i,t}$, grid electricity price λ^{g}_{t}, nodal renewable generation $G^{res}_{i,t}$, nodal power demand level $D^{p}_{i,t}$, and EV charging demand $D^{ev}_{i,t}$.

3.2.2. Action

In the V2H problem, the action $a_{i,t}$ of EV agent i at time step t only captures the power sector and can be generalized as:

$$a_{i,t} = a^{pow}_{i,t}, \forall i \in \mathcal{I}, \ \forall t \in \mathcal{T}, \tag{42}$$

where the continuous scheduling action $a^{pow}_{i,t} \in [-1, 1]$ represents the magnitude of charging (positive) and discharging (negative) power rates of EV agent i as a percentage of its power capacity $[-\overline{P}_i, \overline{P}_i]$.

3.2.3. Reward function

When employing large-scale EVs for V2H problems, the overload issues caused by the power demand of households in the residential areas should be included in the reward function in addition to the charging cost and sufficient travel energy requirement. Note that the power demand of households includes both EV charging characteristics and non-EV loads. Similar to [7,66], the penalty for the transformer

Table 6
Summary of V2G (frequency and voltage regulations).

Ref.	State	Action	Reward	Algorithm	Key features
[83]	Load, voltage, SoC, time	Charging and discharging rate	Minimize voltage deviation, charging and traveling penalties	DDPG, DQN	Propose a human intervention coordinated with DRL to prevent the huge learning loss, realize emergency control, find preferable control policy.
[84]	Local voltage, SoC, active and reactive load	Active and reactive power rate	Limit voltages within acceptable range	Constrained DDPG	Formulate the real-time voltage control problem of EVs as a Markov Game considering both reactive power control and V2G modes of EVs.
[85]	Date, time, load, SoC, weather, traffic flow	Prediction for the boundary condition	Maximize DSO profits	DDPG	Propose an optimal EV charging strategy to maximize DSO profits while satisfying all the physical constraints.
[86]	SoC, RES, active and reactive load, location	Active and reactive power rate	Maximize V2G revenue, minimize non- completion penalty	DDPG	Propose a parameter sharing-based DDPG algorithm to address the coordinated active and reactive power scheduling problem of multiple self-dispatched EVs towards demand-side response and voltage regulations.
[87]	Number of precharged batteries for frequency.	Battery regulation capacity	Maximize charging revenue, frequency support, reduce battery degradation cost	DQN	Schedule the hourly regulation capacity in real time to maximize the battery swapping stations revenue for providing fast frequency regulation services.
[88]	Frequency and voltage deviation, EV active and reactive power output	Active power rate, power angle factor of the charger	Reduce voltage and frequency deviations	DDPG	Aim at the voltage and frequency regulations of microgrid caused by wind disturbance and load fluctuation.

overload can be defined according to each household's contribution to the total power demand as:

$$r^{tf}_{i,t} = \begin{cases} -\frac{P^{dem}_{i,t}}{P^{tf}_{t}}([|P^{tf}_{t}| - \overline{P}^{tf}_{t}])^{2} & \text{if } |P^{tf}_{t}| > \overline{P}^{tf}_{t} \\ 0 & \text{else} \end{cases}, \ \forall i \in \mathcal{I}, \forall t \in \mathcal{T}, \tag{43}$$

where $P^{dem}_{i,t}$ is the household power demand of EV agent i at time step t, $P^{tf}_{t} = \sum P^{dem}_{i,t}$ is its total transformer load, and $\overline{P}^{tf}_{t}$ is the available transformer capacity at time step t.

Combining the charging cost for user satisfaction and the penalty for the overload issue, the reward for EV agent i at time step t is as follows:

$$r^{v2h}_{i,t} = r^{g2v}_{i,t} + \kappa r^{tf}_{i,t}, \ \forall i \in \mathcal{I}, \forall t \in \mathcal{T}, \tag{44}$$

where the trade-off between charging cost and overload penalty can be decided by the weight coefficient κ. In practice, the setting of coefficient κ depends on the users' charging preference [7].

3.2.4. Discussion

Some research limitations and potential solutions to the RL-based V2H problems have been listed and discussed as follows:

3.2.4.1. Privacy issues of evs' cooperation.

- **Research limitations:** The main purpose of this V2H service is to cooperate with electrical devices for load shift through realistic charging and discharging behaviors. In other words, privately owned EVs can work as a controllable load to reduce overloading risk via the onboard or offboard bidirectional charger when they finish their daily journeys and are connected to a home grid [5]. However, potential privacy issues should be carefully addressed during this V2H process, especially when large-scale private EVs are connected and organized for a common target. In this case, SARL methods may not be ideal due to their centralized training and testing framework, compared with MARL methods.
- **Potential solutions:** To address these issues, there have been several papers developing effective MARL algorithms for EV dispatch problems towards V2H service provision, e.g., [7,66–70]. However, current MARL algorithms are still under development with many limitations. If MARL algorithms are implemented under the CTCE and CTDE frameworks, they can still raise privacy issues, and the training procedure may suffer from the curse of dimensionality due to the centralized training procedure, which is impractical in large-scale EV applications. MARL algorithms implemented under the DTDE framework may suffer from severe instability issues, leading to slow convergence and even divergence.
  To this end, there has been some work developing MARL algorithms based on parameter sharing (PS) framework (e.g., [8,58, 92]) to solve the problem, which produces a coordinating strategy for the PEV fleet charging in a distributed manner. Additionally, an attention-based federated RL (FRL) algorithm is proposed in [93] to address the EV charging management problem under a privacy protection mechanism, which can allow all EVs to share the parameters (e.g., the weights of the actor and critic networks) of a single policy. However, similar to CTDE, these approaches still suffer from the curse of dimensionality with regard to the need to incorporate all agents' local observations and actions to estimate the Q-value function, and additionally, the privacy violation still persists.
  Overall, it is still very challenging to design a MARL algorithm that can avoid the privacy issue and the curse of dimensionality.

3.2.4.2. Influence of communication networks.

- **Research limitations:** the physical setting of V2H normally corresponds to small areas, which requires a fast response from EV owners when a potential overloading risk occurs. As such, the information and communication networks in these local areas (e.g., the home area network and the local area network) are critical to provide timely information sharing between EV owners [5]. The delay in signal transmission may severely influence the ability of EVs to provide V2H service. However, there is no research investigating the influence of communication networks on the RL environment setup. Further research into V2H service provision in a time-triggered manner will be conducted in the future.

- **Potential solutions:** To capture the influence of signal transmission delays on EV cooperative dispatch problems, a co-optimization paradigm including both communication systems and power systems is a necessity, which can enhance the collaboration capabilities of large-scale EVs via optimized signal transmission efficiency and energy consumption. It is worth noting that, when communication networks are integrated, power systems are inevitably exposed to cyber-attacks, and the risk of information leakage increases with an increase in communication traffic and distance, which leads to the requirements for blockchain and cyber security technologies [94].

3.3. RL for V2G services

The increasing penetration of renewable energies (e.g., PVs and WTs) into power systems poses operational challenges due to their inherent uncertainty and intermittent nature [95], resulting in a high demand for various ancillary services. As a broad concept, ancillary services can include a wide range of services with various time scales (e.g., seconds, minutes, hours, and even longer) and different perspectives (e.g., demand–supply balance, frequency/voltage regulation, operating reserve, carbon intensity service, and resilience control) [96–98]. As one of the demand-side technologies, EVs have been widely applied in current power systems for ancillary service provision due to their significant advantages in both mobility and flexibility compared to traditional DERs (e.g., flexible demand and energy storage) [99–101]. However, it is worth noting that the large-scale deployment of EV fleets also introduces further challenges to efficient and stable system operations due to the complexity of capturing both power and transport networks. As a result, it is urgent to develop an effective distributed control algorithm for these large-scale and small-size decentralized EVs to exploit their mobility and flexibility in various V2G service provisions.

To address these challenges, RL algorithms have been applied to various V2G problems for real-time energy arbitrage, as summarized in Tables 5 and 6. Under an RL setup, large-scale EVs can better coordinate with each other for effective dispatch behaviors against various uncertainties, e.g., EV departure/arrival time and SOC levels. The general state, action, and reward function used in existing work are detailed as follows:

3.3.1. State

Since there are many different kinds of ancillary services that EV can provide, the most commonly-used state or observations are summarized as follows:

$$o_{i,t} = [t, \Delta_{i,t}, S_{i,t}, \lambda_t^g, G_{i,t}^{res}, D_{i,t}^p, \lambda_t^c, \Delta_{i,t}^F, \Delta_{i,t}^V, D_{i,t}^q], \forall i \in \mathcal{I}, \forall t \in T, \tag{45}$$

where the first six observation features in $o_{i,t}$ are the same as the previous two parts, allowing EV agents to provide grid balance services. Once the carbon intensity price signals λ_t^c are observed, the carbon service to the national grid can be achieved. Finally, the other ancillary services such as frequency and voltage regulations can be provided by EV agents when grid frequency deviation $\Delta_{i,t}^F$, voltage deviation $\Delta_{i,t}^V$, and nodal reactive power demand $D_{i,t}^q$ are available.

3.3.2. Action

The action $a_{i,t}$ of an EV agent i at time step t can be generalized as:

$$a_{i,t} = [a_{i,t}^{pow,p}, a_{i,t}^{pow,q}], \ \forall i \in \mathcal{I}, \forall t \in T, \tag{46}$$

where the continuous power scheduling actions $a_{i,t}^{pow,p} \in [-1, 1]$ and $a_{i,t}^{pow,q} \in [-1, 1]$ represent the output rates of active and reactive power of EV agent i as a percentage of its power capacity $[-\overline{P}_i, \overline{P}_i]$, but are also limited by its apparent power capacity $\overline{S}_i^{pow}$. Furthermore, only the action $a_{i,t}^{pow,p}$ is required for EVs if the goal is to provide energy balance and carbon services. Otherwise, both active and reactive power actions are required, e.g., when EVs are used to provide frequency or voltage regulation services.

3.3.3. Reward function

When EVs choose to provide grid balance service, the reward for discharging power to the grid can be written as:

$$r_{i,t}^e = \lambda_t^g P_{i,t}^d, \ \forall i \in \mathcal{I}, \forall t \in T, \tag{47}$$

where $P_{i,t}^d$ refers to the quantity of power discharge of EV agent i at time step t.

Similarly, when EVs choose to provide carbon intensity service, the reward for discharging power to the grid can be written as:

$$r_{i,t}^c = \lambda_t^c P_{i,t}^d, \ \forall i \in \mathcal{I}, \forall t \in T, \tag{48}$$

where the carbon price λ_t^c observed by EV agent i at time step t in response to carbon intensity signals (gCO$_2$/kWh) can be forecasted and estimated in real time by the National Grid's Carbon Intensity API [102]. This carbon intensity forecast includes CO$_2$ emissions from all large metered power stations, interconnector imports, transmission, and distribution losses, and also accounts for the national electricity demand, embedded wind, and solar generation.

Additionally, EVs can also be used to provide resilience service (e.g., load restoration), as studied in [8]. In this case, the reward function can be written as:

$$r_{i,t}^r = \lambda_i^{ls} P_{i,t}^d, \ \forall i \in \mathcal{I}, \forall t \in T, \tag{49}$$

where λ_i^{ls} is the load shedding cost of EV agent i, which refers to the load priority, e.g., essential load and non-essential load. Note that a resilient power system should mainly focus on the restoration of essential loads (e.g., medical facilities and trading centers), given the large disruptions caused by extreme events [103].

Furthermore, EVs can also be used to provide frequency and voltage regulation services when a certain level of disturbances occur in power systems that may cause system instability issues. Using the voltage regulation service as an example, the reward function can be written as:

$$r_{i,t}^v = \begin{cases} -(\underline{V} - V_{i,t}) & \text{if } V_{i,t} < \underline{V} \\ 0 & \text{if } V_{i,t} \in [\underline{V}, \overline{V}] \\ -(V_{i,t} - \overline{V}) & \text{if } V_{i,t} > \overline{V} \end{cases} \ \forall i \in \mathcal{I}, \forall t \in T, \tag{50}$$

where $\overline{V}$ and $\underline{V}$ are the nodal voltage upper and lower limits (e.g., 0.95 p.u. and 1.05 p.u.), respectively. It is worth noting that the reward function $r_{i,t}^f$ for the frequency regulation service can be carried out in a similar manner, as discussed in [88].

Finally, the reward function for the V2G problem of EV agent i at time step t can be summarized as

$$r_{i,t}^{v2g} = r_{i,t}^{g2v} + r_{i,t}^e + r_{i,t}^c + r_{i,t}^r + +\beta_1 r_{i,t}^v + \beta_2 r_{i,t}^f, \ \forall i \in \mathcal{I}, \forall t \in T, \tag{51}$$

where $r_{i,t}^{g2v}$ is the G2V reward designed in (40). To further balance the importance of service provisions between voltage regulation and frequency regulation, two weight coefficients β_1 and β_2 are introduced. This is because they are not directly related to the EVs' monetary reward.

3.3.4. Discussion

Some research limitations and potential solutions to the RL-based V2G problems have been listed and discussed as follows:

3.3.4.1. Ancillary services related to climate change.

- **Research limitations:** As shown in Tables 5 and 6, there have been plenty of studies on EVs' energy arbitrage problems for energy imbalance services. However, there is not much research on other types of ancillary services. Specifically, only one Ref. [8] applied MARL to EV dispatch problems for carbon intensity services and resilience enhancement, where both normal operation and emergency operation are considered to fully reveal the advantages of EVs in reducing carbon emissions and improving load survivability. It is worth noting that many countries have passed regulations to restrict fossil fuel consumption of traditional vehicles, promising a low-carbon future, which has led to a rapid increase in the use of EVs [2,35].

Additionally, high-impact and low-probability (HILP) events have happened more frequently in recent years, partly because of the rapid climate change [103]. According to [104], seven of ten major storms during the last four decades have happened in the last decade, while each event caused a huge economic loss (over $1 billion). As one type of mobile power source, EVs can be called up in a short time and appropriately deployed for resilience enhancement in the presence of HILP events [105–108]. As such, these two types of services have become important in recent years and deserve further investigation.

- **Potential solutions**: To address these challenges caused by climate change and HILP events, it is necessary to develop a comprehensive framework for EV dispatch problems, fully revealing the advantages of EVs in proving carbon intensity services and enhancing resilience. In this context, RL algorithms incorporating different modules (e.g., normal mode for carbon intensity and emergency mode for resilience [8]) can be developed to appropriately address these two issues. Going further, multi-objective RL methods (e.g., pareto reinforcement learning) [89] may also be an option for EV dispatch problems towards multi-service provisions.

3.3.4.2. Ancillary services related to frequency and voltage regulations.

- **Research limitations**: There are not many papers focusing on developing RL algorithms for EV dispatch problems towards frequency and voltage regulations. In fact, with the V2G technology and inverter-based interface at local charging stations, both the active and reactive power of a charging EV can be controlled within certain charging and capacity limits, and adjusted in a real-time fashion following specific control signals for voltage and frequency regulations [109]. Going further, most existing research on voltage and frequency regulations is based on SARL algorithms, which may not be suitable for large-scale EV dispatch problems due to privacy concerns; thus, MARL algorithms should be further developed to bridge the gap in this area.
- **Potential solutions**: To apply MARL algorithms for EV dispatch problems towards frequency and voltage regulations, privacy perseverance must be one of the most important concerns, leading to the requirements for PS-based or attention-based RL frameworks [86]. Additionally, frequency and voltage instabilities can directly influence the secure operations of power systems that are normally regarded as critical infrastructure in modern societies. In this context, safe RL methods [90] that are capable of handling all the physical constraints of power systems can be developed to ensure secure system operations when large-scale EVs are integrated into power systems for frequency and voltage regulations.

4. Challenges and future perspectives

This section presents the critical challenges and future research directions of using RL algorithms in EV dispatch problems, including five aspects: (1) real-world data availability, (2) detailed RL environment setup, (3) safety and robustness of trained policies, (4) efficient RL training performance, and (5) real-world RL deployment. Note that all these five aspects are directly associated with the reviewed RL-based EV dispatch problems, including G2V, V2H, and V2G. The applications of RL-based algorithms on other research areas are not involved. Because of this narrow and specific research focus, the key discussions that follow can be very concise and straightforward. These aspects are highlighted in Fig. 8 and have been discussed thoroughly as follows:

4.1. Real-world data availability

Capturing various uncertainties associated with state features is becoming a necessity for current EV dispatch problems due to the highly uncertain environment, e.g., time-varying electricity price signals, demand profiles, renewable generation, real-time traffic statuses, etc. According to the existing work, RL algorithms essentially learn from interactions with the environment, implying that they amass a large amount of knowledge on various datasets that can reflect the characteristics of state features. In addition, real-world scenarios are normally characterized by more complex and chaotic data (e.g., incorrectly formatted, corrupted, or incomplete data within a dataset), which leads to the requirement for data cleaning and mining procedures that can be used as pre-treatment techniques to improve the data quality [110] for RL-based EV dispatch problems. In detail, various types of measurement data can be collected from advanced metering infrastructure (AMI) using smart meters and communication networks [111], which are preprocessed via data mining techniques and then benefit the application of data-driven RL algorithms in many ways, e.g., improving data efficiency and determining effective observations [112].

An EV battery operation should capture realistic models and parameters, such as power and energy capacities, charging and discharging efficiencies, energy consumption on the road, and even the degradation cost associated with the number of charging and discharging cycles, from the perspective of an EV agent. More importantly, the dynamic models of the battery energy transition are not only related to the charging and discharging efficiencies, but are also influenced by many other factors, such as battery materials and vehicle weights [113]. To this end, future research will investigate more realistic models of EV batteries for the experiment environment. Furthermore, studies will focus on large-scale heterogeneous EV fleets considering the collective behavior of drivers and fast charging abilities, which can better reflect real-world EV dispatch characteristics. All the aspects need to be realized by a huge amount of real-world data, which further increases the complexity of data sampling. When large-scale datasets are unavailable, data availability will become an issue for the application of various RL problems to EV dispatch problems.

Furthermore, there may not be enough data reflecting real-world EV dispatch characteristics that is available for the RL training process. In this case, a potential solution is to construct training samples or generate virtual samples from the limited data of existing system operations to boost the data availability [114]. It would also be interesting to explore the incorporation of recurrent neural network (RNN) based layers (such as long short-term memory (LSTM) or gated recurrent units (GRU)) to deal with time-series data (e.g., electricity price and renewable generation) for performance enhancement [115].

4.2. Detailed RL environment setup

In the existing work on EV dispatch problems, the RL environment is mainly built on the power sector, including various electricity price signals, renewable generation, load information, etc. However, there are not many papers that consider a more realistic power network model (e.g., voltage limits, power flows, etc.) in the RL setup, which can lead to unrealistic EV charging and discharging behaviors. Furthermore, the power network environment can be extended to capture important operating conditions of power distribution systems, such as dynamic voltage regulation, frequency response, reactive power support, etc. Finally, the impact of local EV flexibility on the national level and wholesale market can be also captured through a tri-level model [116,117].

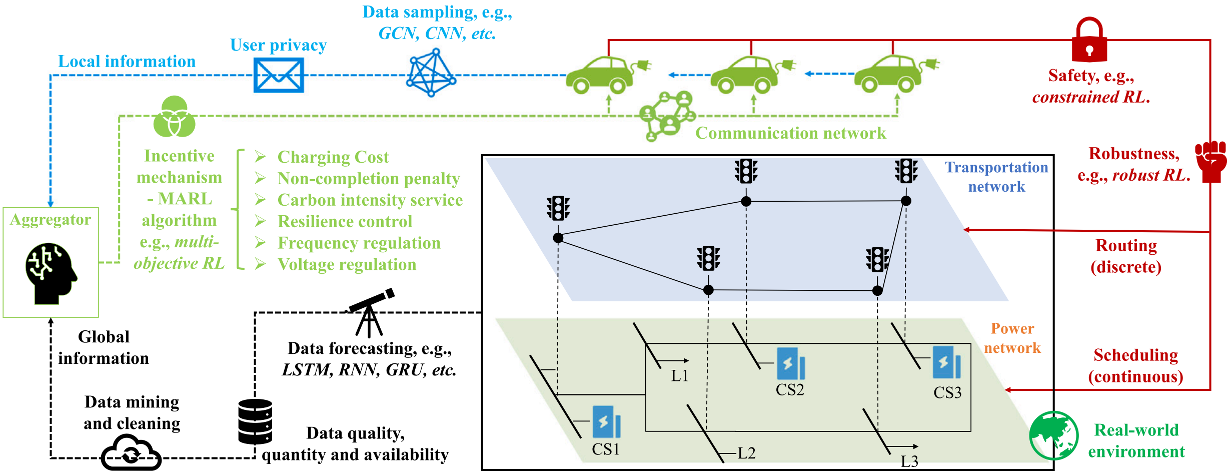


Fig. 8. Challenges and future perspectives of reinforcement learning algorithms applied to electric vehicle dispatch problems in power systems.

Except for the power network, the RL environment can also include the transport network with real-time traffic information to capture detailed EV routing characteristics, as suggested in [8,81]. Additional sources of flexibility (e.g., heat pumps in heat networks, gas boilers in gas networks, etc.) can be considered in the RL environment setup to further reflect the complexity of real-world integrated energy systems. It is worth noting, however, that reliable communication networks capable of capturing transmission delays must be considered in the RL setup to ensure more realistic energy transactions between different sectors. The follow-up work may focus on in-depth analysis and research in these directions and may even add corresponding hardware circuit experiments or semi-physical simulation experiments.

The main goal of EV agents is still to seek economic benefits. As such, appropriate pricing mechanisms in different energy markets can also be included in the RL setup to provide incentives for EV owners that can increase their willingness to contribute to V2G service provisions. Future and ongoing work can focus on investigating appropriate market mechanisms and user behaviors to improve the coordination between a heterogeneous set of EV fleets for providing multiple ancillary services [81].

4.3. Safety and robustness of trained policies

Existing work on EV dispatch problems mainly focuses on the detailed EV routing and scheduling characteristics and ignores the influence of physical constraints related to stability properties (e.g., voltage limits and network congestion) on optimal results, which can lead to unsafe EV dispatch behaviors and even destroy the secure operation of power systems [84]. It should be noted that power systems are critical infrastructures in modern societies. Therefore, it is critical to ensure that the EV dispatch behaviors do not cause the power system operation to violate critical physical constraints or cause any instability issues [10]. However, it is difficult to verify whether a trained policy is safe or whether the generated actions can ensure zero-constraint violations since the training process of DNN for conventional RL algorithms is an unconstrained optimization problem that ignores system physical constraints [75].

In order to address such practical issues, it is a common practice to formulate the constraint violation as a penalty term and add it to the reward function [118,119]. However, as discussed in Section 3.1.4, this can make the reward function very complicated, and it may be challenging to select the suitable values for penalty weighting factors. Furthermore, when a large number of constraints need to be penalized, this approach cannot guarantee that the control policy always leads to safe power system operations.

An alternative way is to formulate the investigated problems as a constrained MDP (CMDP) to handle physical constraints of power system operations, e.g., modeling these as penalty functions, chance constraints, or budget constraints [115,120], even though such schemes usually lead to conservative results and may not be able to handle complex and highly dynamic system environments. Additionally, adversarial RL and robust RL are two advanced algorithms to deal with parametric uncertainties, data errors, and mismatches between simulators and real-world systems, which can also benefit secure power system operation, as discussed in [10]. However, since reliability and security issues are crucial in power systems, the safety of the power system operation must be guaranteed all the time (even during the initial exploration of the RL training process), which is unattainable in these methods. Finally, authors in [121] combine MARL algorithm with binary integer programming (BLP) and propose a Value Decomposition Network (VDN) to solve the real-time scheduling problem in battery charging stations with the joint action constraints. As such, further efforts are required to develop more safe and robust RL algorithms to ensure zero-constraint violations during the test and even training process.

4.4. Efficient RL training performance

Sample efficiency (or data efficiency) in RL [122] means that the algorithm can make better use of the samples collected, resulting in faster policy learning. Using the same number of training samples (e.g., counted by time steps in MDP), a sample-efficient RL algorithm can perform better in terms of the learning curve or final results than other sample-inefficient RL algorithms. However, for existing RL algorithms, it takes more than hundreds or thousands of samples to gradually learn the optimal policies. This poses a key question in RL: how to design more effective RL algorithms for agents that can learn faster with fewer samples? The importance of this issue is mainly because real-time or real-world agents often require a certain amount of time and energy consumption in interacting with their environment. Learning from expert demonstrations [123] is a potential way to improve the sampling efficiency of RL algorithms. This idea requires an expert to provide the training samples with high reward values, which falls under the category of Imitation Learning [124]. It attempts not only to imitate the expert's action choices but also to learn a generalization policy that solves unseen states. The combination of imitation learning and RL is also a promising area of research that can be applied to EV dispatch problems to alleviate the issue of low sampling efficiency in conventional RL algorithms.

When large-scale EVs are integrated, more advanced techniques such as graph attention networks (GAN) [125] can be used to strengthen the cooperation among EV agents. Furthermore, new EV agents may be willing to participate in this collaboration scheme; in this case, knowledge from expert EVs can be shared with the newcomer, for example, by using transfer learning [126] to share knowledge among agents, which avoids the time-consuming task of re-training RL algorithms. In addition, using convolutional neural networks (CNN) and graph neural network (GNN) for function approximation might be more useful under specific circumstances than normal neural networks [127], especially when there is a need to represent agent observations as a matrix, which deserves further investigation. To model the large-scale EV scheduling problem, it is also possible to use the mean-field MARL algorithm [128], which has been successfully deployed to the large-scale peer-to-peer energy trading problem for multi-energy prosumers in a local energy market.

Finally, RL algorithms normally involve many hyper-parameters that need to be carefully selected. Thus, future research should focus on evaluating the sensitivity of hyper-parameters to policy quality. In any RL algorithm, the discount factor γ greatly influences the training performance. A larger γ expecting a long-term return may cause instability of the RL policy; while a smaller γ learning a myopia policy may converge to the local optimum. As a result, selecting a suitable value of γ becomes important for the RL policy. The discount factor γ is typically used to calculate the effective time range for single-step action selection: $1+\gamma+\gamma^2+\cdots=1/(1-\gamma)$. For example, for $\gamma=0.99$, we can usually disregard the reward after 100 time steps.

To further improve the training performance, RL with a distributed structure may be investigated, where multi-threaded parallel computing can provide a reliable algorithm basis for promoting adaptability [129]. Compared to centralized structures, distributed RL approaches do not require any data sharing at the central server, which can significantly reduce the RL learning time by training each agent using its own data [53]. However, under the MARL setting, some RL agents may not be able to acquire enough training data, resulting in an overfitting problem and thereby yielding inaccurate policies [66]. In this context, FRL is a promising solution to tackle the overfitting issue by periodically updating local neural network models, while preventing data privacy leakage at the server [82]. Going further, attention-based information sharing mechanisms can be incorporated into the FRL framework for more efficient coordination of RL agents with enhanced scalability and privacy protection [93].

4.5. Real-world RL deployment

There have been many studies developing advanced MARL methods for EV dispatch problems for different purposes. Nevertheless, most existing MARL methods are trained and tested on small-scale systems with several EV agents, which can be impractical in real-world scenarios. The main reason for this scalability issue is that as the number of agents increases, the state and action spaces expand dramatically, resulting in the curse of dimensionality, as discussed in Section 2.2.6. Several advanced techniques such as parameter sharing and abstracted critic network have been applied to multi-agent EV dispatch problems to deal with this issue [8], while they are still under development with many limitations.

The ultimate goal of any RL algorithm is real-world deployment [130]. In the existing studies, there is no research deploying and testing their trained RL policies in real-world applications. As such, future work is required to further validate the applicability of various RL methods to real-world EV dispatch problems. For instance, as a start, well-trained RL methods can be extended to cover a broader range of parameter settings (e.g., use of finer decision timeslots), and then be validated with hardware circuit experiments or semi-physical simulation experiments, which can improve the safety and interoperability of RL algorithms. After the comprehensive validation, RL algorithms may be able to be deployed in industrial applications and conduct real-world operational tests.

5. Conclusion

This work first provides a comprehensive review of the applications of various RL algorithms to EV dispatch problems. The key components of MDP including agents, environment, state, action, policy function, reward, and state transition function are presented in detail, while classical SARL algorithms (Q-learning, DQN, PG, DDPG, PPO, and SAC) and advanced frameworks for MARL algorithms (CTCE, DTDE, and CTDE) are compared and discussed thoroughly in this work. Three different aspects of the applications of RL algorithms to EV dispatch problems are summarized in a comprehensive way, including G2V, V2H, and V2G. Finally, several key challenges and future research directions are discussed from five different perspectives: real-world data availability; detailed RL environment setup; safety and robustness of trained policies; efficient RL training performance; and real-world RL deployment.

To summarize, many practical problems still remain unsolved when employing advanced RL algorithms for EV dispatch problems, even though there have been plenty of studies in this field. Specifically, real-world data quality and availability are two of the main issues that influence the training performance of RL algorithms, while the insufficient safety and robustness issues of existing RL algorithms in ensuring zero-constraint violations limit their applications in real-world scenarios due to the high-security requirements of modern power systems. Furthermore, the balance between economic EV dispatch results and user privacy issues has not been well addressed in the existing studies. Even though many works have developed different types of MARL methods to simulate EV dispatch cases, the scalability, reliability, and stability of MARL algorithms should be further investigated before deploying them for real-world applications.

CRediT authorship contribution statement

Dawei Qiu: Methodology, Data curation, Validation, Formal analysis, Writing – original draft, Writing – review & editing. **Yi Wang:** Methodology, Data curation, Validation, Formal analysis, Writing – original draft, Writing – review & editing. **Weiqi Hua:** Methodology, Writing – original draft, Writing – review & editing. **Goran Strbac:** Conceptualization, Project administration, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

UK-China project “Technology Transformation to Support Flexible and Resilient Local Energy Systems (grant No EP/T021780/1)” funded by UK RI.

References

[1] Lopion P, Markewitz P, Robinius M, Stolten D. A review of current challenges and trends in energy systems modeling. Renew Sustain Energy Rev 2018;96:156–66.

[2] Dowling P. The impact of climate change on the European energy system. Energy Policy 2013;60:406–17.

[3] Carmichael R. Behaviour change, public engagement and net zero, a report for the committee on climate change. Centre for Energy Policy and Technology (ICEPT) and Centre for Environmental; 2019.

[4] Moustakas K, Loizidou M, Rehan M, Nizami A. A review of recent developments in renewable and sustainable energy systems: Key challenges and future perspective. Renew Sustain Energy Rev 2020;119:109418.

[5] Liu C, Chau K, Wu D, Gao S. Opportunities and challenges of vehicle-to-home, vehicle-to-vehicle, and vehicle-to-grid technologies. Proc IEEE 2013;101(11):2409–27.

[6] Tushar MHK, Zeineddine AW, Assi C. Demand-side management by regulating charging and discharging of the EV, ESS, and utilizing renewable energy. IEEE Trans Ind Inform 2018;14(1):117–26. http://dx.doi.org/10.1109/TII.2017.2755465.

[7] Yan L, Chen X, Chen Y, Wen J. A cooperative charging control strategy for electric vehicles based on multi-agent deep reinforcement learning. IEEE Trans Ind Inf 2022.

[8] Qiu D, Wang Y, Zhang T, Sun M, Strbac G. Hybrid multi-agent reinforcement learning for electric vehicle resilience control towards a low-carbon transition. IEEE Trans Ind Inf 2022.

[9] Sutton RS, Barto AG. Reinforcement learning: an introduction. MIT Press; 2018.

[10] Chen X, Qu G, Tang Y, Low S, Li N. Reinforcement learning for selective key applications in power systems: Recent advances and future challenges. IEEE Trans Smart Grid 2022.

[11] Vázquez-Canteli JR, Nagy Z. Reinforcement learning for demand response: A review of algorithms and modeling techniques. Appl Energy 2019;235:1072–89.

[12] Yang T, Zhao L, Li W, Zomaya AY. Reinforcement learning in sustainable energy and electric systems: A survey. Annu Rev Control 2020;49:145–63.

[13] Perera A, Kamalaruban P. Applications of reinforcement learning in energy systems. Renew Sustain Energy Rev 2021;137:110618.

[14] Wang Z, Hong T. Reinforcement learning for building controls: The opportunities and challenges. Appl Energy 2020;269:115036.

[15] Mason K, Grijalva S. A review of reinforcement learning for autonomous building energy management. Comput Electr Eng 2019;78:300–12.

[16] Shaukat N, Khan B, Ali S, Mehmood C, Khan J, Farid U, et al. A survey on electric vehicle transportation within smart grid system. Renew Sustain Energy Rev 2018;81:1329–49.

[17] Yang Z, Li K, Foley A. Computational scheduling methods for integrating plug-in electric vehicles with power systems: A review. Renew Sustain Energy Rev 2015;51:396–416.

[18] Peng M, Liu L, Jiang C. A review on the economic dispatch and risk management of the large-scale plug-in electric vehicles (PHEVs)-penetrated power systems. Renew Sustain Energy Rev 2012;16(3):1508–15.

[19] Bhatti G, Mohan H, Singh RR. Towards the future of smart electric vehicles: Digital twin technology. Renew Sustain Energy Rev 2021;141:110801.

[20] Venegas FG, Petit M, Perez Y. Active integration of electric vehicles into distribution grids: Barriers and frameworks for flexibility services. Renew Sustain Energy Rev 2021;145:111060.

[21] Watkins CJ, Dayan P. Q-learning. Mach Learn 1992;8(3):279–92.

[22] Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, et al. Human-level control through deep reinforcement learning. Nature 2015;518(7540):529–33.

[23] Riedmiller M. Neural fitted Q iteration–first experiences with a data efficient neural reinforcement learning method. In: European conference on machine learning. Springer; 2005, p. 317–28.

[24] Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. 2017, arXiv preprint arXiv:1707.06347.

[25] Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, et al. Continuous control with deep reinforcement learning. 2015, arXiv preprint arXiv:1509.02971.

[26] Fujimoto S, Hoof H, Meger D. Addressing function approximation error in actor-critic methods. In: International Conference on Machine Learning. PMLR; 2018, p. 1587–96.

[27] Haarnoja T, Zhou A, Abbeel P, Levine S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: International conference on machine learning. PMLR; 2018, p. 1861–70.

[28] Tesauro G, et al. Temporal difference learning and TD-gammon. Commun ACM 1995;38(3):58–68.

[29] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 2014;15(1):1929–58.

[30] Sutton RS, McAllester D, Singh S, Mansour Y. Policy gradient methods for reinforcement learning with function approximation. Adv Neural Inf Process Syst 1999;12.

[31] Silver D, Lever G, Heess N, Degris T, Wierstra D, Riedmiller M. Deterministic policy gradient algorithms. In: International conference on machine learning. PMLR; 2014, p. 387–95.

[32] Lowe R, Wu YI, Tamar A, Harb J, Pieter Abbeel O, Mordatch I. Multi-agent actor-critic for mixed cooperative-competitive environments. Adv Neural Inf Process Syst 2017;30.

[33] Terry JK, Grammel N, Hari A, Santos L. Parameter sharing is surprisingly useful for multi-agent deep reinforcement learning. 2020.

[34] Wang X-C, Klemeš JJ, Dong X, Fan W, Xu Z, Wang Y, et al. Air pollution terrain nexus: A review considering energy generation and consumption. Renew Sustain Energy Rev 2019;105:71–85.

[35] Bellocchi S, Klöckner K, Manno M, Noussan M, Vellini M. On the role of electric vehicles towards low-carbon energy systems: Italy and Germany in comparison. Appl Energy 2019;255:113848.

[36] Hulagu S, Celikoglu HB. An electric vehicle routing problem with intermediate nodes for shuttle fleets. IEEE Trans Intell Transp Syst 2020.

[37] Sadeghianpourhamami N, Deleu J, Develder C. Definition and evaluation of model-free coordination of electrical vehicle charging with reinforcement learning. IEEE Trans Smart Grid 2019;11(1):203–14.

[38] Jin J, Xu Y. Optimal policy characterization enhanced actor-critic approach for electric vehicle charging scheduling in a power distribution network. IEEE Trans Smart Grid 2020;12(2):1416–28.

[39] Zhang F, Yang Q, An D. CDDPG: A deep-reinforcement-learning-based approach for electric vehicle charging control. IEEE Internet Things J 2020;8(5):3075–87.

[40] Chiş A, Lundén J, Koivunen V. Reinforcement learning-based plug-in electric vehicle charging with forecasted price. IEEE Trans Veh Technol 2016;66(5):3674–84.

[41] Yang A, Sun H, Zhang X. Deep reinforcement learning strategy for electric vehicle charging considering wind power fluctuation. J Eng Sci Technol Rev 2021;14(3).

[42] Wang R, Chen Z, Xing Q, Zhang Z, Zhang T. A modified rainbow-based deep reinforcement learning method for optimal scheduling of charging station. Sustainability 2022;14(3):1884.

[43] Wang S, Bi S, Zhang YA. Reinforcement learning for real-time pricing and scheduling control in EV charging stations. IEEE Trans Ind Inf 2019;17(2):849–59.

[44] Zhao Z, Lee CKM. Dynamic pricing for EV charging stations: A deep reinforcement learning approach. IEEE Trans Transp Electrif 2022;8(2):2456–68.

[45] Dorokhova M, Martinson Y, Ballif C, Wyrsch N. Deep reinforcement learning control of electric vehicle charging in the presence of photovoltaic generation. Appl Energy 2021;301:117504.

[46] Vandael S, Claessens B, Ernst D, Holvoet T, Deconinck G. Reinforcement learning of heuristic EV fleet charging in a day-ahead electricity market. IEEE Trans Smart Grid 2015;6(4):1795–805.

[47] Qin Z, Liu D, Hua H, Cao J. Privacy preserving load control of residential microgrid via deep reinforcement learning. IEEE Trans Smart Grid 2021;12(5):4079–89.

[48] Qian T, Shao C, Li X, Wang X, Shahidehpour M. Enhanced coordinated operations of electric power and transportation networks via EV charging services. IEEE Trans Smart Grid 2020;11(4):3019–30.

[49] Qian T, Shao C, Wang X, Shahidehpour M. Deep reinforcement learning for EV charging navigation by coordinating smart grid and intelligent transportation system. IEEE Trans Smart Grid 2019;11(2):1714–23.

[50] Zhang C, Liu Y, Wu F, Tang B, Fan W. Effective charging planning based on deep reinforcement learning for electric vehicles. IEEE Trans Intell Transp Syst 2020;22(1):542–54.

[51] Xing Q, Xu Y, Chen Z, Zhang Z, Shi Z. A graph reinforcement learning-based decision-making platform for real-time charging navigation of urban electric vehicles. IEEE Trans Ind Inf 2022.

[52] Xu P, Zhang J, Gao T, Chen S, Wang X, Jiang H, et al. 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 2022;141:108030.

[53] Shin M, Choi D-H, Kim J. Cooperative management for PV/ESS-enabled electric vehicle charging stations: A multiagent deep reinforcement learning approach. IEEE Trans Ind Inf 2019;16(5):3493–503.

[54] Shi J, Gao Y, Wang W, Yu N, Ioannou PA. Operating electric vehicle fleet for ride-hailing services with reinforcement learning. IEEE Trans Intell Transp Syst 2019;21(11):4822–34.

[55] Liang Y, Ding Z, Ding T, Lee W-J. Mobility-aware charging scheduling for shared on-demand electric vehicle fleet using deep reinforcement learning. IEEE Trans Smart Grid 2020;12(2):1380–93.

[56] Zhang Y, Zhang Z, Yang Q, An D, Li D, Li C. EV charging bidding by multi-DQN reinforcement learning in electricity auction market. Neurocomputing 2020;397:404–14.

[57] Lu Y, Liang Y, Ding Z, Wu Q, Ding T, Lee W-J. Deep reinforcement learning-based charging pricing for autonomous mobility-on-demand system. IEEE Trans Smart Grid 2022;13(2):1412–26.

[58] Qian T, Shao C, Li X, Wang X, Chen Z, Shahidehpour M. Multi-agent deep reinforcement learning method for EV charging station game. IEEE Trans Power Syst 2021;37(3):1682–94.

[59] Tuchnitz F, Ebell N, Schlund J, Pruckner M. Development and evaluation of a smart charging strategy for an electric vehicle fleet based on reinforcement learning. Appl Energy 2021;285:116382.

[60] Jiang Y, Ye Q, Sun B, Wu Y, Tsang DH. Data-driven coordinated charging for electric vehicles with continuous charging rates: A deep policy gradient approach. IEEE Internet Things J 2021.

[61] Bertolini A, Martins MS, Vieira SM, Sousa JM. Power output optimization of electric vehicles smart charging hubs using deep reinforcement learning. Expert Syst Appl 2022;116995.

[62] Lee J, Lee E, Kim J. Electric vehicle charging and discharging algorithm based on reinforcement learning with data-driven approach in dynamic pricing scheme. Energies 2020;13(8):1950.

[63] Liu D, Wang W, Wang L, Jia H, Shi M. Dynamic pricing strategy of electric vehicle aggregators based on DDPG reinforcement learning algorithm. IEEE Access 2021;9:21556–66.

[64] Wang K, Wang H, Yang J, Feng J, Li Y, Zhang S, et al. Electric vehicle clusters scheduling strategy considering real-time electricity prices based on deep reinforcement learning. Energy Rep 2022;8:695–703.

[65] Alqahtani M, Hu M. Dynamic energy scheduling and routing of multiple electric vehicles using deep reinforcement learning. Energy 2022;244:122626.

[66] Da Silva FL, Nishida CE, Roijers DM, Costa AHR. Coordination of electric vehicle charging through multiagent reinforcement learning. IEEE Trans Smart Grid 2019;11(3):2347–56.

[67] Jiang C, Jing Z, Cui X, Ji T, Wu Q. Multiple agents and reinforcement learning for modelling charging loads of electric taxis. Appl Energy 2018;222:158–68.

[68] Li S, Hu W, Cao D, Zhang Z, Huang Q, Chen Z, et al. EV charging strategy considering transformer lifetime via evolutionary curriculum learning-based multi-agent deep reinforcement learning. IEEE Trans Smart Grid 2022.

[69] Al Zishan A, Haji MM, Ardakanian O. Adaptive congestion control for electric vehicle charging in the smart grid. IEEE Trans Smart Grid 2021;12(3):2439–49.

[70] Li S, Hu W, Cao D, Zhang Z, Huang Q, Chen Z, et al. A multi-agent deep reinforcement learning-based approach for the optimization of transformer life using coordinated electric vehicles. IEEE Trans Ind Inf 2022.

[71] Wang J, Guo C, Yu C, Liang Y. Virtual power plant containing electric vehicles scheduling strategies based on deep reinforcement learning. Electr Power Syst Res 2022;205:107714.

[72] Tao Y, Qiu J, Lai S. Deep reinforcement learning based bidding strategy for EVAs in local energy market considering information asymmetry. IEEE Trans Ind Inf 2022;18(6):3831–42.

[73] Yan L, Chen X, Zhou J, Chen Y, Wen J. Deep reinforcement learning for continuous electric vehicles charging control with dynamic user behaviors. IEEE Trans Smart Grid 2021;12(6):5124–34.

[74] Qiu D, Ye Y, Papadaskalopoulos D, Strbac G. A deep reinforcement learning method for pricing electric vehicles with discrete charging levels. IEEE Trans Ind Appl 2020;56(5):5901–12.

[75] Li H, Wan Z, He H. Constrained EV charging scheduling based on safe deep reinforcement learning. IEEE Trans Smart Grid 2019;11(3):2427–39.

[76] Wan Z, Li H, He H, Prokhorov D. Model-free real-time EV charging scheduling based on deep reinforcement learning. IEEE Trans Smart Grid 2018;10(5):5246–57.

[77] Lee S, Choi D-H. Energy management of smart home with home appliances, energy storage system and electric vehicle: A hierarchical deep reinforcement learning approach. Sensors 2020;20(7):2157.

[78] Li S, Hu W, Cao D, Dragičević T, Huang Q, Chen Z, et al. Electric vehicle charging management based on deep reinforcement learning. J Mod Power Syst Clean Energy 2021.

[79] Wang F, Gao J, Li M, Zhao L. Autonomous PEV charging scheduling using Dyna-Q reinforcement learning. IEEE Trans Veh Technol 2020;69(11):12609–20.

[80] Gao X, Chan KW, Xia S, Zhang X, Zhang K, Zhou J. A multiagent competitive bidding strategy in a pool-based electricity market with price-maker participants of WPPs and EV aggregators. IEEE Trans Ind Inf 2021;17(11):7256–68.

[81] Qiu D, Wang Y, Sun M, Strbac G. Multi-service provision for electric vehicles in power-transportation networks towards a low-carbon transition: A hierarchical and hybrid multi-agent reinforcement learning approach. Appl Energy 2022;313:118790.

[82] Lee S, Choi D-H. Dynamic pricing and energy management for profit maximization in multiple smart electric vehicle charging stations: A privacy-preserving deep reinforcement learning approach. Appl Energy 2021;304:117754.

[83] Tao Y, Qiu J, Lai S, Zhang X, Wang Y, Wang G. A human-machine reinforcement learning method for cooperative energy management. IEEE Trans Ind Inf 2022;18(5):2974–85.

[84] Sun X, Qiu J. A customized voltage control strategy for electric vehicles in distribution networks with reinforcement learning method. IEEE Trans Ind Inf 2021;17(10):6852–63.

[85] Ding T, Zeng Z, Bai J, Qin B, Yang Y, Shahidehpour M. Optimal electric vehicle charging strategy with Markov decision process and reinforcement learning technique. IEEE Trans Ind Appl 2020;56(5):5811–23.

[86] Wang Y, Qiu D, Strbac G, Gao Z. Coordinated electric vehicle active and reactive power control for active distribution networks. IEEE Trans Ind Inf 2022;1. http://dx.doi.org/10.1109/TII.2022.3169975.

[87] Wang X, Wang J, Liu J. Vehicle to grid frequency regulation capacity optimal scheduling for battery swapping station using deep Q-network. IEEE Trans Ind Inf 2020;17(2):1342–51.

[88] Fan P, Ke S, Kamel S, Yang J, Li Y, Xiao J, et al. A frequency and voltage coordinated control strategy of island microgrid including electric vehicles. Electronics 2021;11(1):17.

[89] Hu X, Zhang Y, Liao X, Liu Z, Wang W, Ghannouchi FM. Dynamic beam hopping method based on multi-objective deep reinforcement learning for next generation satellite broadband systems. IEEE Trans Broadcast 2020;66(3):630–46.

[90] Li H, He H. Learning to operate distribution networks with safe deep reinforcement learning. IEEE Trans Smart Grid 2022.

[91] Zeng P, Li H, He H, Li S. Dynamic energy management of a microgrid using approximate dynamic programming and deep recurrent neural network learning. IEEE Trans Smart Grid 2018;10(4):4435–45.

[92] Qiu D, Ye Y, Papadaskalopoulos D, Strbac G. Scalable coordinated management of peer-to-peer energy trading: A multi-cluster deep reinforcement learning approach. Appl Energy 2021;292:116940.

[93] Chu Y, Wei Z, Fang X, Chen S, Zhou Y. A multiagent federated reinforcement learning approach for plug-in electric vehicle fleet charging coordination in a residential community. IEEE Access 2022;10:98535–48.

[94] Wang Y, Chen C-F, Kong P-Y, Li H, Wen Q. A cyber–physical–social perspective on future smart distribution systems. Proc IEEE 2022.

[95] Zhang Y, Wang J, Li Z. Uncertainty modeling of distributed energy resources: techniques and challenges. Curr Sustain/ Renew Energy Rep 2019;6(2):42–51.

[96] Zhou Y, Wu J, Song G, Long C. Framework design and optimal bidding strategy for ancillary service provision from a peer-to-peer energy trading community. Appl Energy 2020;278:115671.

[97] Ruan G, Wu J, Zhong H, Xia Q, Xie L. Quantitative assessment of US bulk power systems and market operations during the COVID-19 pandemic. Appl Energy 2021;286:116354.

[98] Wang Y, Qiu D, Strbac G. Multi-agent reinforcement learning for electric vehicles joint routing and scheduling strategies. In: 2022 IEEE 25th international conference on intelligent transportation systems. IEEE; 2022, p. 3044–9.

[99] DeForest N, MacDonald JS, Black DR. Day ahead optimization of an electric vehicle fleet providing ancillary services in the Los Angeles air force base vehicle-to-grid demonstration. Appl Energy 2018;210:987–1001.

[100] Shang W-L, Chen J, Bi H, Sui Y, Chen Y, Yu H. Impacts of COVID-19 pandemic on user behaviors and environmental benefits of bike sharing: A big-data analysis. Appl Energy 2021;285:116429.

[101] Ruan G, Wu D, Zheng X, Zhong H, Kang C, Dahleh MA, et al. A cross-domain approach to analyzing the short-run impact of COVID-19 on the US electricity sector. Joule 2020;4(11):2322–37.

[102] National Grid. Carbon intensity API - national data. 2021, URL https://carbonintensity.org.uk/.

[103] Wang Y, Rousis AO, Strbac G. On microgrids and resilience: A comprehensive review on modeling and operational strategies. Renew Sustain Energy Rev 2020;134:110313.

[104] Hussain A, Bui V-H, Kim H-M. Microgrids as a resilience resource and strategies used by microgrids for enhancing resilience. Appl Energy 2019;240:56–72.

[105] Gao H, Chen Y, Mei S, Huang S, Xu Y. Resilience-oriented pre-hurricane resource allocation in distribution systems considering electric buses. Proc IEEE 2017;105(7):1214–33.

[106] Wang Y, Rousis AO, Strbac G. A resilience enhancement strategy for networked microgrids incorporating electricity and transport and utilizing a stochastic hierarchical control approach. Sustain Energy Grids Netw 2021;26:100464.

[107] Wang Y, Qiu D, Strbac G. Multi-agent deep reinforcement learning for resilience-driven routing and scheduling of mobile energy storage systems. Appl Energy 2022;310:118575.

[108] Wang Y, Rousis AO, Strbac G. Resilience-driven optimal sizing and pre-positioning of mobile energy storage systems in decentralized networked microgrids. Appl Energy 2022;305:117921.

[109] Sbordone D, Bertini I, Di Pietra B, Falvo MC, Genovese A, Martirano L. EV fast charging stations and energy storage technologies: A real implementation in the smart micro grid paradigm. Electr Power Syst Res 2015;120:96–108.

[110] Hand DJ. Principles of data mining. Drug Saf 2007;30(7):621–2.

[111] Mohassel RR, Fung A, Mohammadi F, Raahemifar K. A survey on advanced metering infrastructure. Int J Electr Power Energy Syst 2014;63:473–84.

[112] Wang Q, Li F, Tang Y, Xu Y. Integrating model-driven and data-driven methods for power system frequency stability assessment and control. IEEE Trans Power Syst 2019;34(6):4557–68.

[113] Balali Y, Stegen S. Review of energy storage systems for vehicles based on technology, environmental impacts, and costs. Renew Sustain Energy Rev 2021;135:110185.

[114] Xu H, Domínguez-García AD, Sauer PW. Optimal tap setting of voltage regulation transformers using batch reinforcement learning. IEEE Trans Power Syst 2019;35(3):1990–2001.

[115] Qiu D, Dong Z, Zhang X, Wang Y, Strbac G. Safe reinforcement learning for real-time automatic control in a smart energy-hub. Appl Energy 2022;309:118403.

[116] Qiu D, Papadaskalopoulos D, Ye Y, Strbac G. Investigating the effects of demand flexibility on electricity retailers' business through a tri-level optimisation model. IET Gener Transm Distrib 2020;14(9):1739–50.

[117] Qiu D, Dong Z, Ruan G, Zhong H, Strbac G, Kang C. Strategic retail pricing and demand bidding of retailers in electricity market: A data-driven chance-constrained programming. Adv Appl Energy 2022;7:100100.

[118] Lei L, Tan Y, Dahlenburg G, Xiang W, Zheng K. Dynamic energy dispatch based on deep reinforcement learning in IoT-driven smart isolated microgrids. IEEE Internet Things J 2021;8(10):7938–53.

[119] Guo C, Wang X, Zheng Y, Zhang F. Real-time optimal energy management of microgrid with uncertainties based on deep reinforcement learning. Energy 2022;238:121873.

[120] Zhang Q, Dehghanpour K, Wang Z, Qiu F, Zhao D. Multi-agent safe policy learning for power management of networked microgrids. IEEE Trans Smart Grid 2021;12(2):1048–62.

[121] Liang Y, Ding Z, Zhao T, Lee W-J. Real-time operation management for battery swapping-charging system via multi-agent deep reinforcement learning. IEEE Trans Smart Grid 2022.

[122] Yu Y. Towards sample efficient reinforcement learning. In: IJCAI. 2018, p. 5739–43.

[123] Ramírez J, Yu W, Perrusquía A. Model-free reinforcement learning from expert demonstrations: a survey. Artif Intell Rev 2022;55(4):3213–41.

[124] Hussein A, Gaber MM, Elyan E, Jayne C. Imitation learning: A survey of learning methods. ACM Comput Surv 2017;50(2):1–35.

[125] Zhang W, Liu H, Han J, Ge Y, Xiong H. Multi-agent graph convolutional reinforcement learning for dynamic electric vehicle charging pricing. In: Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining. 2022, p. 2471–81.

[126] Weiss K, Khoshgoftaar TM, Wang D. A survey of transfer learning. J Big Data 2016;3(1):1–40.

[127] Wu T, Scaglione A, Arnold D. Complex-value spatio-temporal graph convolutional neural networks and its applications to electric power systems AI. 2022, arXiv preprint arXiv:2208.08485.

[128] Qiu D, Wang J, Dong Z, Wang Y, Strbac G. Mean-field multi-agent reinforcement learning for peer-to-peer multi-energy trading. IEEE Trans Power Syst 2022.

[129] Tang X, Chen J, Liu T, Qin Y, Cao D. Distributed deep reinforcement learning-based energy and emission management strategy for hybrid electric vehicles. IEEE Trans Veh Technol 2021;70(10):9922–34.

[130] Luo W, Sun P, Zhong F, Liu W, Zhang T, Wang Y. End-to-end active object tracking and its real-world deployment via reinforcement learning. IEEE Trans Pattern Anal Mach Intell 2019;42(6):1317–32.