https://dx.doi.org/10.1016/j.jobe.2023.106774

Operational optimization for the grid-connected residential photovoltaic-battery system using model-based reinforcement learning

Yang Xu[a, b], Weijun Gao[a, b, *], Yanxue Li[a, c] , Fu Xiao[c]

a. Innovation Institute for Sustainable Maritime Architecture Research and Technology, Qingdao University of Technology, Fushun Road 11, Qingdao, 266033, China;
b. Faculty of Environmental Engineering, The University of Kitakyushu, Kitakyushu, 808-0135, Japan
c. Department of Building Environment and Energy Engineering, The Hong Kong Polytechnic University, 100872, Hong Kong, China
* Corresponding Author: Weijun Gao; gaoweijun@me.com

Abstract

The development of distributed photovoltaic and energy storage devices has created challenges for energy management systems due to uncertainty and mismatch between local generation and residents' energy demand. Reinforcement learning is gaining attention as a control algorithm, but traditional model-free RL has data quality and quantity limitations for energy management applications. Therefore, this study proposed a model-based deep RL method to optimize the operation control of the energy storage system by taking the measured dataset of an actual existing building in Japan as the research object. With an optimization goal of reducing the microgrid's energy cost and ensuring the PV self-consumption ratio, we designed a new reward function for these goals. We took the benchmark strategy currently used by the target building's energy management system as the baseline model in the experiment. We applied four advanced RL algorithms (PPO, DQN, DDPG, and TD3) to optimize the baseline model. The results show that the proposed RL design can better achieve the two optimization objectives of minimizing energy cost and maximizing the PV self-consumption ratio. Among them, the TD3 algorithm presented the best performance. Compared with the baseline model, its annual energy cost can be reduced by 17.82%, and the photovoltaic self-consumption ratio can be increased by 0.86%. In addition, the model-based RL method proposed in this paper can provide a better energy management strategy with the training set of only one and a half years of measured data, which proves that it has a high potential for practical application.

Keywords: Deep reinforcement learning, Operational optimization, Photovoltaic battery systems, Actor-critic algorithms

Nomenclature

Abbreviations

RES	renewable energy sources
PV	photovoltaic
HMES	household multi-energy system
RTP	real-time electricity price
ESS	home energy management system
MPC	model predictive control
ML	machine learning
MPC	model predictive control
MDP	markov decision process
RL	reinforcement learning
DQN	deep q-networks
DDPG	deep deterministic policy gradient
TD3	Twin Delayed Deep Deterministic Policy Gradients
DDQN	Double deep q-networks
SOC	State of charge

Environment Parameter Symbols

E	ESS power, Wh
P	power from demand or generation, Wh
C	energy price, Yen/Wh
η	Efficiency
r	PV self-consumption ratio

Reinforcement Learning Symbols

S	state
A	action
P	state transition probability
R	reward
γ	discount factor
$Q(s,a)$	critic action-value estimate
$\mu(s)$	actor policy action
θ^Q	critic network weights
θ^μ	actor network weights

1. Introduction

In recent years, owing to the rapid development of industrialization and urbanization, the global energy demand has risen sharply, which has brought severe challenges to mitigate climate change. Since building energy consumption accounts for about 40% of global energy consumption[1], increasing the proportion of renewable energy sources(RES) to reduce building energy consumption has become a research hotspot[2]. Since photovoltaic(PV) technology has the advantages of excellent cost and convenient deployment, making it one of the most widely used RES[3]. For example, the Japanese government has introduced a series of incentive policies for applying RES[4]. Consequently, more and more households in Japan are opting for the household multi-energy system (HMES)[5], which integrates electricity, natural gas, and renewable energy sources (such as photovoltaic and wind power) as energy sources. As a bidirectional grid-connected energy system, HMES can meet multiple load demands of users and sell excess renewable energy to the grid, reducing household

energy payment costs[6]. Therefore, the HMES integrating RES undoubtedly has significant research value and application potential.

However, due to the multiple uncertainties in the application of the HMES, the energy scheduling of the system faces significant challenges. Firstly, renewable energy production is greatly affected by environmental factors (such as weather conditions)and has strong intermittency and uncertainty. Secondly, with the development of the electricity market, many countries have adopted the real-time electricity price (RTP)[7], which is also highly uncertain due to the fluctuations of the electricity futures trading price. Third, for residential customers, the differences in living habits and rapid electrification will also lead to the uncertainty of electricity demand. The energy storage system (ESS) is an effective approach to deal with these uncertainties[8]. The ESS can not only effectively alleviate the instability caused by the fluctuation of renewable energy but also optimize the economy of the energy system according to the dynamic information of energy prices, and the grid-connected residential photovoltaic-battery system based on HMES has become Japan's fastest-growing renewable energy technology[6]. It should be noted that although the ESS has the above advantages, it also increases the system cost and the complexity of system optimization[9].

To further improve the economy of ESS and the utilization of renewable energy, intelligent ESS has attracted increasing attention. Intelligent ESS enables more efficient energy management by introducing control systems that formulate optimal control strategies considering renewable energy production, electricity demand, and RTP[10–12]. Most current residential intelligent ESS systems use classical control methods, such as proportional, integral, and derivative controllers or rule-based controllers. However, these controllers cannot predict the many uncertainties in the system because they do not include domain-specific knowledge and cannot use historical data or model predictions. Therefore, traditional control methods can not achieve relatively accurate energy storage control. Model predictive control (MPC) is a popular multi-objective control method, which could formulate these uncertainties as a constrained optimization problem[13–15]. MPC is a control strategy that integrates dynamic analysis techniques, allowing it to evaluate time-based scenarios and make more accurate decisions[16–18]. For example, the MPC controller can advance charge or discharge control of ESS based on the forecast of demand, RTP, and renewable energy production to improve renewable energy utilization and save energy costs. However, since the prediction of future data and the setting of constraints are the basis of MPC model implementation, its control effect is greatly affected by the model's prediction accuracy[19]. An accurate prediction model often needs a large amount of training data and careful hyperparameter tuning. This implies that knowledge learned by the MPC model is difficult to transfer between different buildings because the historical data of each residential customer is unique and has different requirements and

characteristics. Therefore, developing a standard MPC model for different residential customers is a severe challenge.

Due to the increasing popularity of Machine Learning (ML) methods, Markov decision process (MDP) theory-based reinforcement learning (RL) provides an effective solution to solve the operational optimization problem of building energy systems. Compared with MPC, the RL model does not require complex and accurate plant modeling. It can make the RL agent interact with the environment through training data, select the action that maximizes the cumulative reward, and then make an optimal decision, which makes it possible to make a standard model for different buildings[20].

1.1 Related Work

Due to the above advantages of the RL method, research on applying RL to the operation optimization of building energy systems has increased significantly over the past decade[21]. As the most classical value-based RL algorithm, Q-learning is the earliest RL method proposed and applied in energy systems due to its model-free and easy-to-evaluate strategy. For example, Waldemar et al. proposed a Q-learning-based method to optimize the non-stationary environment and non-linear storage characteristics of the storage-integrated PV system and verified through simulation experiments that it could reduce the cost of energy purchased on a real-time basis to a minimum[22]. Authors in[23] proposed a model-free Q-learning method that makes optimal control decisions for HVAC and window systems to minimize both energy consumption and thermal discomfort. The work in [24]uses the Q-learning method to optimize a residential RES and reduce energy consumption by improving the utilization rate of renewable energy. However, the Q-learning method records the optimization knowledge using the Q-value table. When the system's state or action space is too large, it will lead to the curse of dimensionality, which limits the application of Q-learning methods.

With the development of deep learning(DL) technology, deep RL (DRL) has been proposed to solve the above problems. Combining the powerful non-linear fitting ability of deep neural networks with the excellent decision-making ability of RL, the DRL can overcome some previously tricky issues, such as decision problems in continuous action spaces. Harrold et al.[25]adopted the Rainbow Deep Q-Networks(DQN) method to control batteries in a microgrid for arbitrage. Experimental results show that this method is superior to the actor-critic and linear programming methods, which could effectively carry out arbitrage according to demand, PV generation, and RTP. Authors in [26]proposed a DRL method based on the deep deterministic policy gradient (DDPG) that can minimize the energy cost of smart home energy systems via controlling Heating, Ventilation, and

Air Conditioning (HVAC) and ESS. Li et al.[27]proposed an end-to-end cooling control algorithm (CCA) based on the DDPG. The results show that the proposed CCA can achieve up to 11% cooling cost reduction on the EnergyPlus simulation platform compared with a manually configured baseline control algorithm. Mocanu et al.[28]compared the operation optimization effects of DQN and DPG algorithms on building energy systems. The experimental results show that the DPG with continuous action space is superior to the DQN with discrete action space, which could reduce the building operation cost by 27.4% and the peak load by 26.3%. Y Du et al.[29] adopted the DDPG algorithm to generate the optimal HVAC control strategy with the minimum energy consumption cost while maintaining the users' comfort. The simulation results show that compared with DQN, the control strategy based on DDPG can reduce the energy consumption cost by 15% and the comfort violation by 79%.

While the works mentioned above have contributed to the applications of RL technologies in building energy systems scheduling, there are still two limitations of these approaches. First, the optimization of ESS is mainly focused on a single optimization objective, such as the system's economy. Specifically, it uses ESS to arbitrage under RTP fluctuation while ignoring the local consumption of renewable energy, which contradicts the original intention of improving the renewable energy penetration level of the grid. Second, learning control policies using RL methods require enormous amounts of data. Most of the works mentioned above used infinite simulated data generated by building simulators (such as EnergyPlus) or a large amount of actual data (over three years), which leads to time-consuming training. Model-based RL (MB-RL) is one of the methods to overcome this problem. MBRL can use the domain knowledge of the model to learn the optimal control policy in a data-driven way more effectively. Heeyun et al. [30] established a battery energy consumption model based on the domain knowledge of vehicle powertrain for RL training. The simulation results show that compared with the dynamic programming result, the performance of MBRL reaches 93.8%. Authors in[31] proposed a model-based A3C algorithm to realize the strategic bidding for wind energy. The simulation results show that the strategy generated by MB-A3C is superior to other model-free or model-based RL algorithms.

1.2 Contributions

Based on the above-reviewed work, the contributions of this study can be summarized as follows:

- We adopted an existing grid-connected residential photovoltaic-battery system’s data as the research object. We used an RL-based approach to optimize the system's operation,

including reducing energy costs while maintaining the renewable energy self-consumption ratio within a predetermined range. Therefore, we designed a new reward function to achieve these goals and proved its effectiveness through experiments.

- A model-based Twin Delayed Deep Deterministic Policy Gradients(MB-TD3) algorithm is developed for the operational optimization of residential ESS. The advantage of this method is that domain knowledge is used in RL model configuration to improve data utilization, the exploration scope of the agent is reduced, and the data utilization rate is improved. In addition, the RL agent is also used to explore the systems that are difficult to model, giving full play to the advantages of the model-free method. Detailed case analysis and comparison of four advanced RL algorithms are presented, where the baseline model uses the strategy used in actual buildings. We evaluated the efficiency of these algorithms in terms of energy cost and renewable energy self-consumption ratio and proved that the efficiency of MB-TD3 is optimal.
- We showed that all the MB-RL algorithms could reliably ensure that the renewable energy self-consumption ratio is higher than the baseline model while reducing the electricity purchase cost. This emphasizes these model-based RL algorithms' ability to learn optimal control policies with fewer datasets, providing valuable insights for real-world implementations.

2. Methodology

This section will first explore the fundamentals and classification of the RL and then introduce the specific algorithms used in the case studies later in the paper.

2.1 Reinforcement learning

RL, as a branch of machine learning, is a computational method that can solve sequential decision problems[32,33]. All the RL problems can be defined as MDP, which represents the process by which an agent guides its behavior by obtaining rewards from interaction with the environment. Formally, an MDP is a five-tuple $(S,\ A,\ P,\ R,\ R)$, where:

- S is the state space, which represents the available information that the RL agents use to make decisions.
- A is the action space, which means the RL agents make different decisions when interacting with the environment.

- P is the state-transition probability, which describes the probability distribution of going from state s to state s' when action a is taken.
- R is the reward (or cost) function, usually the objective function in a control problem.
- r is the discount factor. The discount factor is used to overcome the feedback delay in the interaction between the agent and the environment. By discounting the rewards for multiple steps, the sum of the accumulated rewards for numerous steps in the future can be obtained. Then the short-term optimization objective and long-term optimization objective can be balanced.

RL aims to find the optimal policy π through the Markov decision process, which refers to the mapping of states to actions[34]. Specifically, the mapping is constructed by the state-value function $v_\pi(s)$ and the action-value functions $q_\pi(s,a)$, which formula is as follows:

$$v_\pi(s) = E_\pi\left[\sum_{k=v}^{\infty} r^k R_{t+k+1} | S_t = s\right] \quad (1)$$

$$q_\pi(s,a) = E_\pi\left[\sum_{k=v}^{\infty} r^k R_{t+k+1} | S_t = s, A_t = a\right] \quad (2)$$

Through Eq.1 and Eq.2, we can obtain the Behrman equation of the state-action value function:

$$q_\pi(s,a) = E_\pi[R_{t+1} + rq(S_t, A_t) | S_t = s, A_t = a] \quad (3)$$

If the optimal state-action value function $q^*(s,a)$ is known, the optimal policy can be determined by directly maximizing it:

$$\pi_*(a) = \begin{cases} 1 & if\ a = \underset{a \in A}{argmax} q^*(s,a) \\ 0 & otherwise \end{cases} \quad (4)$$

The above summarized the basic principles of RL. Next, we will discuss the classification of RL algorithms. The overview of the most popular algorithms is summarized in Table 1. According to the different action selection strategies, RL algorithms can be divided into two branches: value-based algorithms and policy-based algorithms. The value-based algorithm calculates the expectation of reward through the potential reward as the basis for selecting actions. And the policy-based algorithm trains a probability distribution through policy sampling and enhances the probability of the desired action with a high return value[32]. Therefore, value-based algorithms can only be used for discrete action space, while policy-based algorithms have more advantages in continuous action space control. Currently, the most popular actor-critic method combines the benefits of these two branches. Specifically, the actor-network will take actions based on the probability distribution of policies, and the critic-network will give the value of actions to the actions, which makes it more convenient for the latter to deal with continuous control. Therefore, this paper focuses on them.

The RL algorithm can be off-policy or on-policy according to the interaction between the RL agent and the environment. For the off-policy method, the agent can learn by interacting with the environment in person or through accumulated experience (such as experience replay or replay

buffer mechanism)[35]. In contrast, for the on-policy method, the agent can only interact with the environment to update the network. As the research object of this study is the measurement data collected by the actual HMES, the amount of data is limited, and the data collection is slow, so we would prefer to choose the off-policy method because they are more sample-efficient. In contrast, the on-policy method is more suitable for scenarios where data is generated using simulators.

Table 1 Common properties of the popular RL algorithms.

Algorithm	Type	Data usage	Action space
DQN	value-based	Off-policy	Discrete
DPG	policy-based	Off-policy	Continuous
DDPG	actor-critic	Off-policy	Continuous
TD3	actor-critic	Off-policy	Continuous
TRPO	policy-based	On-policy	Discrete/Continuous
PPO	actor-critic	On-policy	Discrete/Continuous

2.2 Deep Q-Networks (DQN)

DQN is a value-based RL algorithm that uses a deep neural network (Q-network) to fit Q values. It overcomes the dimension disaster in the traditional Q-learning algorithm. The Q-network estimates the Q value of each discrete action, and $\varepsilon - greedy$ strategy is used to select the action with the highest Q value. In addition, The DQN adds the experience replay mechanism in the training process, which allows the DQN to randomly extract a batch of historical training data from the buffer for gradient descent of the network. This mechanism prevents the training data from being highly temporal correlated, improving the model's efficiency. The update rule of the DQN algorithm is as follows:

$$Q(s_t, a_t; \theta_t) = Q(s_t, a_t; \theta_t) + \alpha\left[r_i + \gamma max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta') - Q(s_t, a_t; \theta_t)\right] \quad (5)$$

Where θ_t means the parameters of the evaluation network, and θ'means the parameters of the target network. DQN estimates the new Q value more accurately by establishing these two independent neural networks.

2.3 Deep deterministic policy gradient (DDPG)

It can be seen from Eq.5 that there is a calculation to find the maximum value when the DQN network is updated. For continuous action space, this maximization operation is impossible. Therefore, the DQN can only handle a finite discrete action space. DDPG algorithm is proposed to solve this problem[36]. Based on the DPG (deterministic policy gradient) algorithm, DDPG integrates the advantages of the DQN (such as experience replay mechanism and independent target

network) and enables it to deal with the continuous action space by introducing the actor-critic framework.

The learning process of DDPG is shown in Fig.1. As we can see, the DDPG consists of two DNNs: online actor network $\mu(s|\theta^{\mu})$ and online critic network $Q(s,a|\theta^{Q})$, target actor $\mu'(s|\theta^{\mu})$ and target critic networks $Q'(s,a|\theta^{Q})$ are also used to stabilize learning, which has the same structure and initial parameters as the online network. It should be noted that the weights of all the above networks are fixed in training and updated at the end of each step. DDPG also borrows from DQN's experience replay mechanism, which is one of the common strategies used in on-policy methods. It can learn by sampling previous transitions from limited data, thus effectively improving the sample efficiency. It is beneficial for the training environment with relatively small data samples.

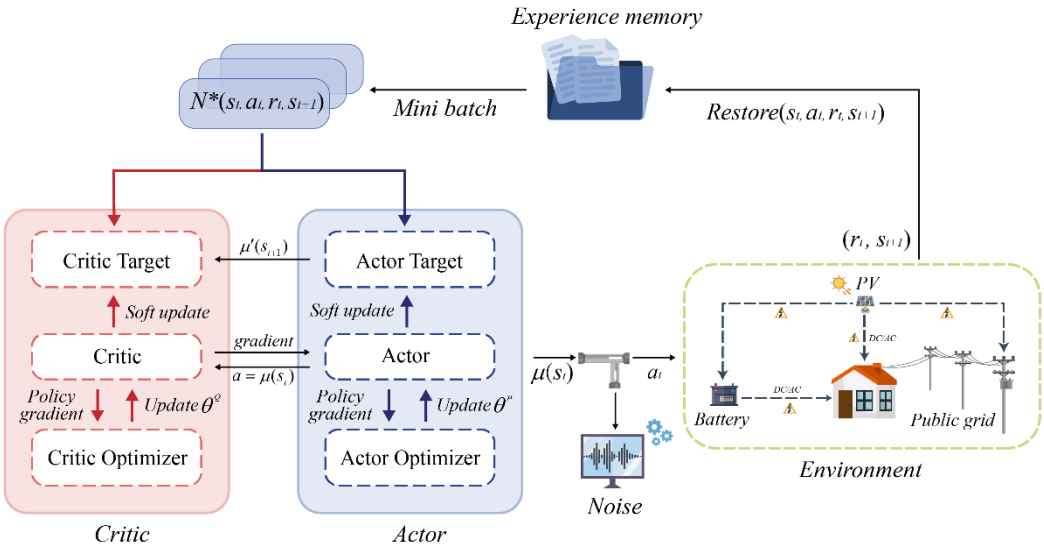

Fig. 1. Basic schematic of the DDPG

When the agent begins to explore, the action selection is performed by passing the current state and random noise x_t through the actor network:

$$a_t = \mu(s_t|\theta^{\mu}) + x_t \tag{6}$$

Once the environment has executed the a_t, the agent will observe the reward r_t and the new state s_{t+1}, and then store the transition data tuples (s_t, a_t, r_t, s_{t+1}) in an experience replay buffer. Mini-batch sampling is performed using a replay buffer for training. The critic and target critic networks will evaluate the target value y_i by observing s_t and a_t, then update the critic network by minimizing the loss function L[36], as shown in Eq.7 and Eq.8:

$$y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'}) \tag{7}$$

$$L = \frac{1}{N}\Sigma_i(y_i - Q(s_i, a_i|\theta^{Q}))^2 \tag{8}$$

Next, the agent will calculate the policy gradient of the local actor network and update parameters through gradient ascent using the deterministic policy gradient[36]:

$$\nabla_{\theta^\mu}\mu|_{s_t} \approx \frac{1}{N}\sum_i \nabla_a Q(s,a|\theta^Q)|_{s=s_t,a=\mu(s_t)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s_t} \quad (9)$$

At the end of each step, the agent updates the parameters of the target actor and critic networks by the running average method:

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)|\theta^{Q'} \quad (10)$$

$$\theta^{\mu} \leftarrow \tau\theta^\mu + (1-\tau)|\theta^{\mu'} \quad (11)$$

Where τ is a smoothing parameter by which to maintain the stability of training.

2.4 Twin delayed deep deterministic policy gradient (TD3)

Based on the above introduction, we can conclude that DDPG is a variant of DQN to solve the continuous control problem. Therefore, it inherits a series of advantages of DQN but also disadvantages, such as overestimation. Overestimation means the estimated value function is larger than the actual. As seen from Eq.5, when the DQN agent finishes the explore, instead of the actual action of the next interaction, it updates the value function with the action currently considered to have the largest value so that it will overestimate the value of Q.

To solve the overestimation problem, Hasselt first proposed the double Q-Learning method, whose application in DQN is called Double DQN (DDQN)[37]. DDQN uses two value function estimates to perform the best action of the next interaction and target estimate using different value estimates, which effectively optimizes the Q-Value overestimation problem. TD3 aims to solve the overestimation problem for DDPG by using a similar approach with a second critic and target critic pair[38]. Instead of using a specific target network in Eq.7, TD3 uses the minimum of the two critic networks to calculate the target values y_i, as shown in Eq.12:

$$y_i = r_i + \gamma \min_{i=1,2} Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'}) \quad (12)$$

In addition, TD3 reduces the update frequency of the Actor-network. The authors of TD3 found that if the policy is learned after the Q value is stabilized, there will be fewer wrong updates in the Actor-network, which can help stabilize the training[38]. Since TD3 is a minor improvement based on DDPG, its algorithm flow is basically the same as DDPG. Due to space limitations, the algorithm flow of TD3 will not be introduced in this section. The selection of experimental algorithms will be detail discussed in the following sections.

3. Case Study

3.1 Data source

The RL models proposed in this study were verified using the dataset of an actual Japanese house located in the "Jono Zero Carbon Smart Community" in Kitakyushu. The building has an energy system with PV panels (the capacity is 4.18 kW and the conversion efficiency is 19.6%), a storage battery (the capacity is 5.6 kW and the conversion efficiency is 90%), and connected to the public grid. Since the microgrid is a hybrid AC/DC network, inverters are used on the power lines of the battery and PV arrays for AC-DC conversion (The efficiency of the inverter is 95%). Fig. 2 illustrates the concept of the PV-battery system.

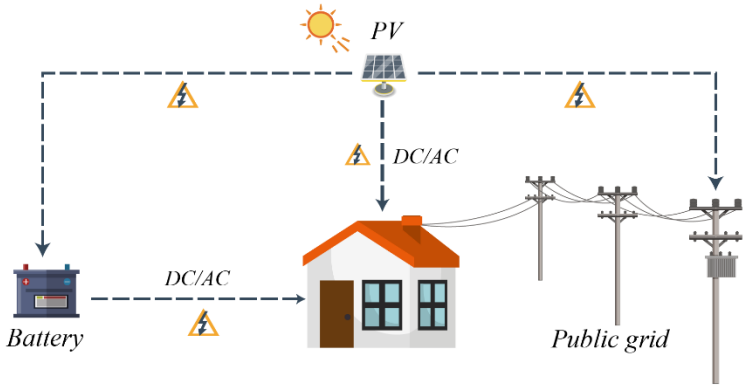

Fig. 2 Structure of the residential PV-battery system

The study dataset was collected by the HEMS implemented in the target house, which includes about thirty months of hourly data from April 1, 2017, to September 30, 2019. These hourly data contain eight components: PV generation (kWh), power demand (kWh), electricity price (Yen), month, hour of day, outdoor temperature, illumination intensity, and humidity. It should be noted that since the real-time electricity price is not implemented in Japan yet, the RTP data adopted in this experiment is simulated. Considering the strong correlation between the fluctuation of RTP and the price of electricity futures, this study uses the electricity futures price published by JEPX (Japan Electric Power Exchange) to multiply by a fixed parameter, making it the same order of magnitude as the current ladder electricity price, and then simulates the real-time electricity price.

The essential characteristics of the dataset are always the basis for experimental design. Fig.3 is the overview of the target dataset. As shown in Fig.3(a), load, PV, and RTP have strong seasonal characteristics, so evaluating the model over a short test set is not comprehensive.

To overcome this, we took one year's data as the test set and divided it into three periods: cooling season, heating season, and transition season, to evaluate the model's performance separately. It can be seen from Fig.3 (b) that the distribution of PV generation and RTP has evident periodicity. For example, the peak of RTP usually occurs between 17:00 and 21:00, which is not coincide with the peak period of PV generation. It also indicates a vast optimization space for energy storage systems. In addition, it should be noted that since the nighttime electricity price of the ladder electricity price is low, the heat pump of this house is set to operate at night, so the mean load fluctuation is slight.

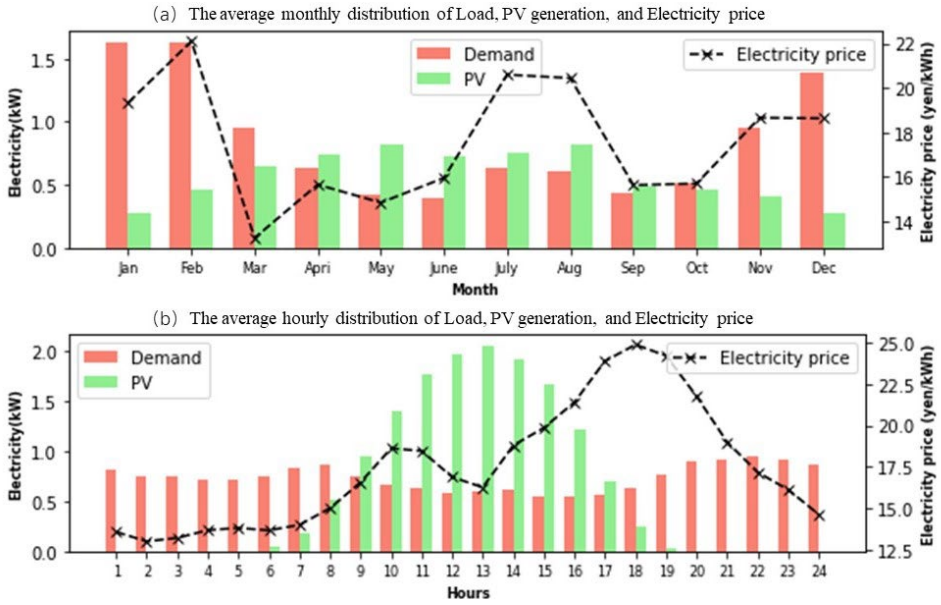

Fig. 3 Overview of the dataset: (a)The average monthly distribution of Load, PV generation, and RTP, (b) The average hourly distribution of Load, PV generation, and RTP

3.2 Baseline control model

The baseline control model of the ESS is mainly based on the power balance formula and the battery capacity constraints, which can be expressed as:

$$P_t^{gird} = P_t^{demand} - P_t^{PV} + P_t^{battery} \quad (13)$$

$$E_{min}^{battery} \leq E_t^{battery} \leq E_{max}^{battery} \quad (14)$$

Where P_t^{gird} is the amount of electricity purchased or sold to the public grid at the time t; P_t^{demand} is the electricity demand at time t; P_t^{PV} is the PV generation at time T; $P_t^{battery}$ is the charge or discharge amount of the battery at time t. $P_t^{battery}$ >0 refers to the charge capacity,

$P_t^{battery}$<0 refers to the discharge capacity. $E_{max}^{battery}$ denotes the maximum battery capacity and $E_{min}^{battery}$ denotes the minimum battery capacity.

In ESS's existing control logic, the battery works only when some fixed conditions are met. Algorithm 1 is the pseudocode of the specific control logic. This control logic avoids the arbitrage of renewable energy using batteries and reduces the energy loss due to battery efficiency, which is as follows:

- When the PV generation is greater than the demand and the battery capacity is less than the $E_{max}^{battery}$, the battery charges and the remaining PV generation is sold to the public grid for revenue. Where η_{cha} denotes the charge efficiency.
- When the PV generation is less than the demand and the battery capacity is greater than the $E_{min}^{battery}$, the battery discharges, and the insufficient power will be purchased from the public grid. Where $\eta_{disscha}$ denotes the discharge efficiency.
- If the above two conditions are not met, the battery will not perform any action.

Algorithm 1 Baseline Control Step

if $P_t^{PV}>P_t^{demand}$ and $E_t^{battery}<E_{max}^{battery}$ do

  $E_{t+1}^{battery}=E_t^{battery}+\eta_{cha}*P_t^{battery}*\Delta t$

end if

elif $P_t^{PV}<P_t^{demand}$ and $E_t^{battery}>E_{min}^{battery}$ do

  $E_{t+1}^{battery}=E_t^{battery}+\frac{P_t^{battery}*\Delta t}{\eta_{disscha}}$

end elif

else $E_{t+1}^{battery}=E_t^{battery}$

end else

The model-based RL method adopted in this study aims to optimize the baseline model using the strategies learned from the data rather than learning a new set of scheduling rules. It means all the proposed RL models will also interact with the environment under the rule of the Baseline control model. That is, the battery performs the action selected by the agent only after the agent determines whether the above conditions for charging and discharging are met. In this model-based RL approach, the agent can use the known rules of the baseline control model for fast and efficient learning, avoiding many unnecessary exploration actions, such as exceeding the battery capacity constraints, frequent selling PV generation to the public grid in pursuit of arbitrage, or other idle behaviors. It means that we can limit and narrow the exploration scope of agents according to the baseline model, thus reducing the number of trials and errors of agents and improving the utilization of training samples.

3.3 Model-based RL application

3.3.1 State space

The state observations S are the values the agents obtain when selecting actions. The state space in this study mainly consists of four parts:

- Energy features: As seen from Fig.3 (b), PV generation, demand, and RTP are periodic in the time series. Therefore, we designed a 24-step (24h) sliding time window (when less than 24 hours, the list filling is 0) to enable the agent to learn their potential rules.
- Time series features: The current hour in the day (X_t^{hour}) and the month (X_t^{month}).
- Environmental features: Outdoor temperature (X_t^{temp}), illumination (X_t^{lux}), and humidity (X_t^{hum}).
- Episode step: The position of the current time step in the entire optimization window (T).

In summary, the state space of proxy observation can be expressed as:

$$S_t=[\ T,\ s_{t-23}^{pv},\dots s_t^{pv},\ s_{t-23}^{demand},\dots s_t^{demand},\ s_{t-23}^{price},\dots s_t^{price},\ X_t^{hour},\ X_t^{month},\ X_t^{temp},\ X_t^{lux},\ X_t^{hum}] \quad (15)$$

To improve the training stability of the RL agent, all the observation values should be normalized in the pre-processing stage, which means that each variable's values should be scaled down to the range of [0,1].

3.3.2 Action space

Since this study's objective is to control the battery continuously, we used the battery control factor to achieve this operation. The battery's actual power was calculated based on the maximum charge and discharge per hour and the battery control factor. The battery control factor ranges from −1 to 1 (the negative sign indicates that the battery is discharging, and the positive sign indicates that the battery is charging), which is also the action space used in this study.

3.3.3 Reward function

Currently, there are two standard reward function design approaches discrete reward function and continuous reward function. As for the discrete reward function, it is easy to converge but contains less information. Conversely, the continuous reward function contains more information, but it is easy to have the problem of sparse rewards, making the training difficult to converge[39]. The optimization objective always determines the design of the reward function. In this study, the

aim of the agents is to reduce the energy cost of the microgrid and ensure that the PV self-consumption ratio is not lower than the baseline model, which can be defined as a multi-objective optimization. The reward function of multi-objective optimization is often designed to consist of multiple parts and constraints. Therefore, we designed the reward function into two parts: economic reward and PV generation consumption reward.

First, the economic reward is calculated by the average cost of electricity imported to or exported from the microgrid. The average cost of electricity during timeslot 0~T is considered:

$$R_{eco} = -(a * \frac{1}{T}\int_{t=0}^{T}\left(P_{gird}(t) * C_{gird}(t) - P_{sell}(t) * C_{sell}\right)\mathrm{d}t) \quad (16)$$

The minus sign indicates that if the average electricity cost is lower, the R_{eco} will be larger. Moreover, a denotes the reward factor, which is a fixed constant that regulates orders of magnitude, by which we can control the order of magnitude of R_{eco} within the range of -10 to 10; $P_{gird}(t)$ denotes the electricity purchased from the public grid by the system at time t, and $C_{gird}(t)$ denotes the real-time electricity price at time t; $P_{sell}(t)$ denotes the electricity sold by the system to the public grid at time t, and C_{sell} denotes the feed-in tariff.

Second, we used a discrete reward function to define the PV generation consumption reward. The calculation of the PV self-consumption ratio is shown in Eq.17, by which we can get the agent's PV self-consumption ratio (r_{RL}) and the baseline model's PV self-consumption ratio ($r_{baseline}$).

$$r = \frac{\int_{t=0}^{T}\left(P_{pv}(t) - P_{sell}(t)\right)\mathrm{d}t}{\int_{t=0}^{T}P_{pv}(t)\,\mathrm{d}t} * 100\% \quad (17)$$

Where $P_{pv}(t)$ denotes the PV generation at time t, and $P_{sell}(t)$ denotes the electricity the microgrid sells to the public grid at time t. If r_{RL} is greater than $r_{baseline}$, then R_{pv} is equal to 1. In contrast, when r_{RL} is less than $r_{baseline}$, R_{pv} is assigned the value -10.

Finally, the sum of the two rewards is the primary reward function, which is as follows:

$$R = R_{eco} + R_{pv} \quad (18)$$

3.4 Experimental setting

3.4.1 Implementation details

Since the ultimate goal of this study is to solve the operational optimization problem of ESS in practical applications, all the cases in this experiment were optimized hour by hour using the actual measured hourly data set. The training set used in this experiment is hourly data from April 1, 2017, to September 30, 2018, and the test data includes hourly data from October 1, 2018, to September 30, 2019. To verify the optimization effects of various RL algorithms under the model-based

framework, we focused on evaluations of the following four algorithms: PPO, DQN, DDPG, and TD3. These four selected algorithms cover all RL technology branches, shown in Section 2.1. In addition, we also added the baseline model for comparison, as shown below:

- M.0: The baseline model adopted in this experiment and its control flow is shown in Section 3.2.
- M.1: M.1 used the PPO algorithm based on the model-based framework proposed in this paper. The PPO algorithm is chosen for comparison since it is a typical on-policy RL method.
- M.2: M.2 used the DQN algorithm based on the model-based framework proposed in this paper. Since DQN is a value-based algorithm that can only handle discrete action space, we use methods that map continuous actions to discrete actions to achieve ESS control. The discrete action space is $[-1, -0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8, 1]$.
- M.3: M.3 used the DDPG algorithm based on the model-based framework proposed in this paper.
- M.4: M.4 used the TD3 algorithm based on the model-based framework proposed in this paper. M.3 and M.4 used the same hyperparameters for comparison.

The deep learning framework adopted in this experiment is PyTorch, in which the environment code was written using the gym framework of OpenAI[40], and the RL algorithms adopted were implemented by the Pytorch version of the Stable Baselines framework[41]. The primary hyperparameters for different algorithm designs are shown in Table 2, and other hyperparameters follow default settings in the stable baseline. Although some algorithms may have a greater reward by fine-tuning the hyperparameters, we prefer to use more default parameters provided by Stable Baselines. Since fine-tuning the hyperparameters is impossible when deployed in a physical residential, we should pay more attention to the generalization ability of the models in practical application.

Table 2 Parameters for different algorithms.

Parameter	PPO	DQN	DDPG	TD3
Activation function	Tanh	Relu	Relu	Relu
Optimiser	Adam	Adam	Adam	Adam
Learning rate	0.0002	0.0002	0.0002	0.0002
Batch size	128	128	128	128
Replay memory capacity	None	1000000	1000000	1000000
Discount factor	0.99	0.99	0.99	0.99
Delay steps in TD3	None	None	None	2

3.4.2 Performance metrics

We will use two metrics to evaluate the algorithm's performance: energy cost and PV self-consumption ratio. The energy cost is calculated by Eq.19. We will evaluate the energy cost of the above five models from annual and monthly dimensions.

$$c = \int_{t=0}^{T} \left(P_{gird}(t) * C_{gird}(t) - P_{sell}(t) * C_{sell} \right) dt \quad (19)$$

As for the PV self-consumption ratio, which calculation was shown in Eq.17. We will also evaluate it from annual and monthly dimensions. The evaluation standard is as long as the baseline model is exceeded as qualified.

4. Result and Discussion

4.1 Training process analysis

Through the callback function provided by Stable Baseline[41], we found that most models generally reach the highest cumulative reward during 50 to 60 episodes of training, and each episode simulates 13199 hours of run optimization. The average episodic rewards of the different algorithms across all 60 episodic can be found in Fig.4. After taking random actions for the first ten episodes for exploration, all agents show similar initial behaviors and begin to gain benefits progressively. After around forty episodes, The average reward growth of all agents starts to slow down and gradually converges. We can see that M.1 fluctuates significantly in the initial stage, which is determined by the nature of its on-policy. Its performance tends to be stable with the increased number of training episodic. It indicates that the training performance of the PPO algorithm based on the on-policy method is weaker than that of other off-policy algorithms on small data samples. Conversely, Although TD3's average reward is close to that of other algorithms in the first ten episodes, it keeps the highest average reward after that. It proves that TD3 performs significantly better than the other algorithms, with a best average reward of more than 0, suggesting that it can extract valuable knowledge from the data more efficiently.

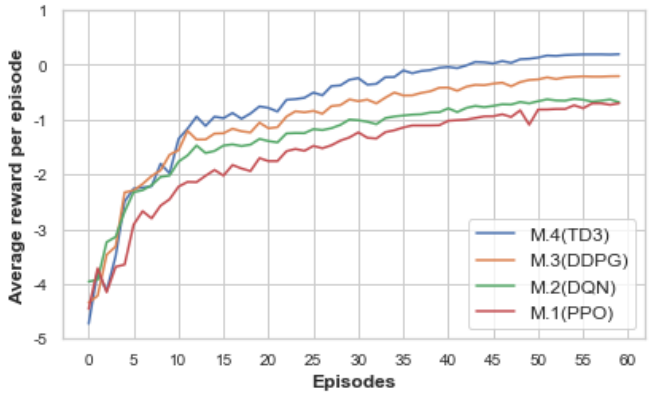

Fig. 4 Average rewards per episodic in the training process

4.2 Analysis of optimization results

This section will summarize the results of the simulation experiment. All the RL models are trained for 50 episodes for fairness and efficiency. We first evaluated all the proposed RL algorithms against the two optimization objectives presented in Section 3.3.3. Then we analyzed the performance of these algorithms in detail through a visual analysis of three typical cases.

4.2.1 Energy cost optimization analysis

First, we compare the annual energy costs of the four algorithms and the baseline model, which is summarized in Table 3. Note that a positive cost represents purchasing power from the public grid, while a negative cost represents the profit from selling PV generation to the public grid. We can see all RL algorithms achieve the optimization of energy cost for the total annual cost. PPO has the worst optimization effect among the four algorithms, with its energy cost reduced by only 4.02% compared to the baseline model. In addition, it did not achieve cost optimization targets in June.

On the contrary, the other three off-policy algorithms have achieved good results, which proves that the off-policy algorithm is more suitable to deal with the practical applications of this scenario with better sample efficiency. Among them, the annual optimization effect of TD3 is the best, which reduces the cost by 17.82% compared with the baseline model. Next came DDPG, close to TD3, with a 15.45% cost reduction relative to the baseline model. It also proves that the actor-critic framework algorithms have advantages in dealing with this scenario.

Table 3 Annual energy cost result and percentage difference against the Baseline model.

Month	M.0(Yen)	M.1(Yen)	M.2(Yen)	M.3(Yen)	M.4(Yen)
Jan	22575.21	21638.04	21049.83	21536.78	21375.14
Feb	19257.42	18631.07	17990.31	18126.05	18151.57
Mar	2424.09	2161.89	2079.55	1999.84	1968.18
Apr	-1276.51	-1329.10	-2021.08	-1996.41	-2013.67
May	-1990.03	-2077.12	-2552.06	-2593.28	-2500.74
Jun	-4016.58	-3949.97	-4806.07	-4706.14	-4831.08
Jul	-348.45	-372.59	-1091.39	-1098.56	-1085.25
Aug	1894.51	1846.35	1449.84	1152.54	1161.03
Sep	1046.97	1011.18	670.38	312.86	280.71
Oct	-610.77	-765.32	-781.51	-1613.40	-1689.44
Nov	9312.58	8941.64	8759.79	7883.95	6980.48
Dec	16151.86	16091.94	15592.44	15463.64	15142.14
Total cost	64420.30	61828.01	56340.03	54467.88	52939.06
VS Baseline		4.02%	12.54%	15.45%	17.82%

Since the test dataset's features fluctuate greatly in different months, as shown in Fig.3, the annual statistics cannot fully reflect the optimization effect of these algorithms. To evaluate the experimental results in more detail, we divided the year into three periods according to the use of HVAC: heating season, cooling season, and transition season. We have calculated the energy cost savings of the four algorithms in these three periods, respectively, and the results are summarized in Table 4.

Table 4 Quarterly cost savings against the Baseline model

	M.1(Yen)	M.2(Yen)	M.3(Yen)	M.4(Yen)
Heating season	1994.37	3904.70	4286.64	5647.73
Cooling season	41.48	2353.69	2915.74	3051.04
Transition season	556.43	1821.88	2750.03	2782.46

We can see that the cost optimization effect of the four algorithms is the best in the heating season. It can be seen from Fig.3a that the energy demand and RTP fluctuation in the heating season are the strongest, while the PV generation is the lowest in the whole year. In this case, the RL agents can pay more attention to the fluctuations of demand and RTP, according to which agents will intelligently choose the time point of charge and discharge to realize the optimization of energy cost. For the cooling season, we can also find in Fig.3a that its average energy demand, RTP, and PV generation are both high, which puts forward higher requirements on the learning ability of agents. The optimization results show that M2, M3, and M4 can learn the rule of feature change well, while M1 fails to achieve this goal. We can also find that the energy demand and PV generation in the transition season are close to that in the cooling season. The only difference is that the RTP in the transition season is relatively low, so the agent only needs to focus on PV and demand schedule

during this period. The performance of M.2, M.3, and M.4 remained stable in this period, while that of M1 also picked up. The reasons for these phenomena are described in detail in section 4.2.3.

4.2.2 PV self-consumption ratio optimization analysis

To ensure the self-consumption ratio of renewable energy and avoid the system using RTP fluctuations for arbitrage, we set the PV self-consumption ratio of the ESS during the optimization period should be higher than the baseline model. We calculated the annual PV self-consumption ratio of the test dataset, and the results are shown in Table 5. It can be seen that all the algorithms have reached the optimization goal. Surprisingly, the PV self-consumption ratio of M.2 was the highest, while M.4 was the least. Since the reward function used in this experiment was composed of energy cost and self-consumption ratio, it showed that different algorithms had different sensitivities to these two parts. In future research, we should try to fine-tune the reward function's weight coefficient a according to different algorithms to obtain a better optimization effect.

Table 5 Annual PV self-consumption ratio results

Algorithm	PV self-consumption ratio
M.0(baseline)	66.80%
M.1(PPO)	68.91%
M.2(DQN)	69.78%
M.3(DDPG)	67.98%
M.4(TD3)	67.66%

The comparison of the PV self-consumption ratio in different months is shown in Fig.5. It can be seen that the photovoltaic self-consumption ratio of M.1 and M.2 is significantly higher than that of M.3 and M.4 in winter. However, their performance is similar in other seasons. It indicates that M.1 and M.2 are more sensitive to the PV self-consumption ratio in winter, thus neglecting the energy cost optimization. The analysis in the previous section showed that the cost optimization effect of M.1 and M.2 was weaker than that of M.3 and M.4, which also confirmed this conclusion.

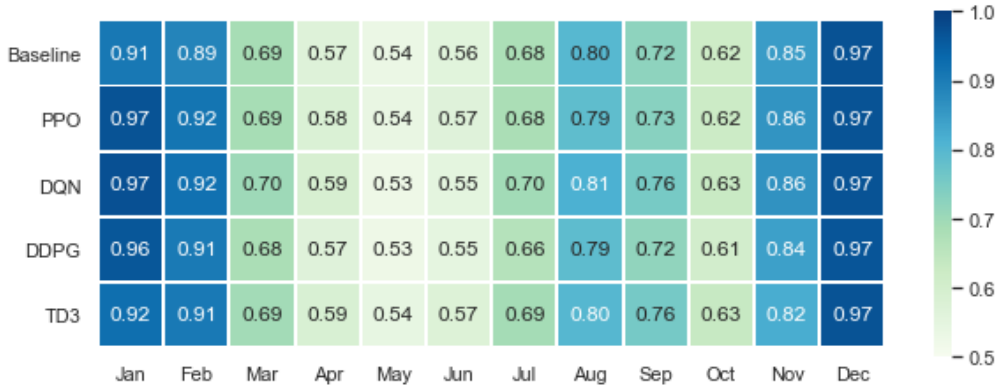

Fig. 5 Thermal map of PV self-consumption ratio

4.2.3 Visualization analysis of optimization results

This section describes the optimization performance of each algorithm in one week to demonstrate specific optimization strategies. We will discuss three typical cases of the heating, cooling, and transition seasons according to the classification standards in Section 4.2.1.

Fig.6 shows the optimization results of different optimization algorithms during one week in the heating season. The left ordinate indicates the state of charge(SOC), the right ordinate indicates the RTP, and the abscissa indicates the hour. For this week's optimization task, we can see that the four algorithms have adjusted based on the baseline model. As seen from Fig.6, all algorithms can predict future demand and RTP trends and adopt the strategy of storing power for the possible price peak in the evening to achieve cost optimization. Both DDPG and TD3 performed well, with optimized results of 285.62(Yen) and 321.78 (Yen), respectively. However, PPO failed to execute the strategy on the second, sixth, and seventh days, thus achieving an optimization result of only 60.2 (Yen). On the contrary, DQN over-implemented this strategy. It chose too low discharge efficiency on the second and fifth days, which resulted in the discharge action missing the peak price, thus achieving a cost optimization result of 190.21 (Yen). This proves that DDPG and TD3 based on the actor-critic framework have a good learning effect on cost optimization.

Fig.7 shows the optimization results of different optimization algorithms during one week in the cooling season. We found that the optimization strategy of each algorithm at this period includes the following two points: (1) When the battery has stored enough power, it will hold it until the evening price peaks occur. (2) When the RTP is low in the morning, the battery will choose to delay charging, and the PV generated during this period will be sold to the public grid for profit. The reason for the appearance of strategy 2 is that PV generation is abundant in summer, and the PV self-consumption ratio can be ensured not to be lower than the baseline model even if the PV sales volume is increased. It can be seen that DDPG and TD3 are more inclined to strategy 1 in the selection of optimization strategy, and they perform well in the selection of discharge time point and slightly increased the sale of PV in the morning, with optimization results of 101.56(Yen) and 167.06(Yen), respectively. In contrast, PPO paid more attention to strategy 2. Although the sale of PV increased, the cost optimization goal was not achieved due to poor selection of discharge action. It can be seen that constrained by the model-based framework, the space for each algorithm to realize cost optimization by increasing the sale of PV is very limited, and proper discharge action selection is the key to ensuring optimization efficiency.

Fig.8 shows the optimization results of different optimization algorithms during one week in the cooling season. The strategies adopted by the algorithms during the transition season are similar

to the cooling season because these two seasons' data distribution is very similar, except for the difference in energy demand. By calculating the optimization result, we found that the optimization effect of TD3 is still the best (805.14 Yen), followed by DDPG (168.33 Yen). It can be seen that the optimization effect of DQN is only a fine-tuning of the baseline model, so it only achieved an optimization structure of 34.57 (Yen). However, PPO failed to optimize again because it incorrectly adjusted the charging rate on the first and fourth days, resulting in insufficient power, thus reducing the utilization of renewable energy.

From the above analysis, it can be seen that the model-based RL methods proposed in this study can provide useful feedback for dynamic control considering the fluctuations in RTP and electricity demand, i.e., they could estimate the value or policy of future states or actions based on current observations, which undoubtedly reflects the characteristics of dynamic analysis techniques. Since the dynamic analysis features are integrated into the model-based RL methods, they have advantages in solving such scenario problems.

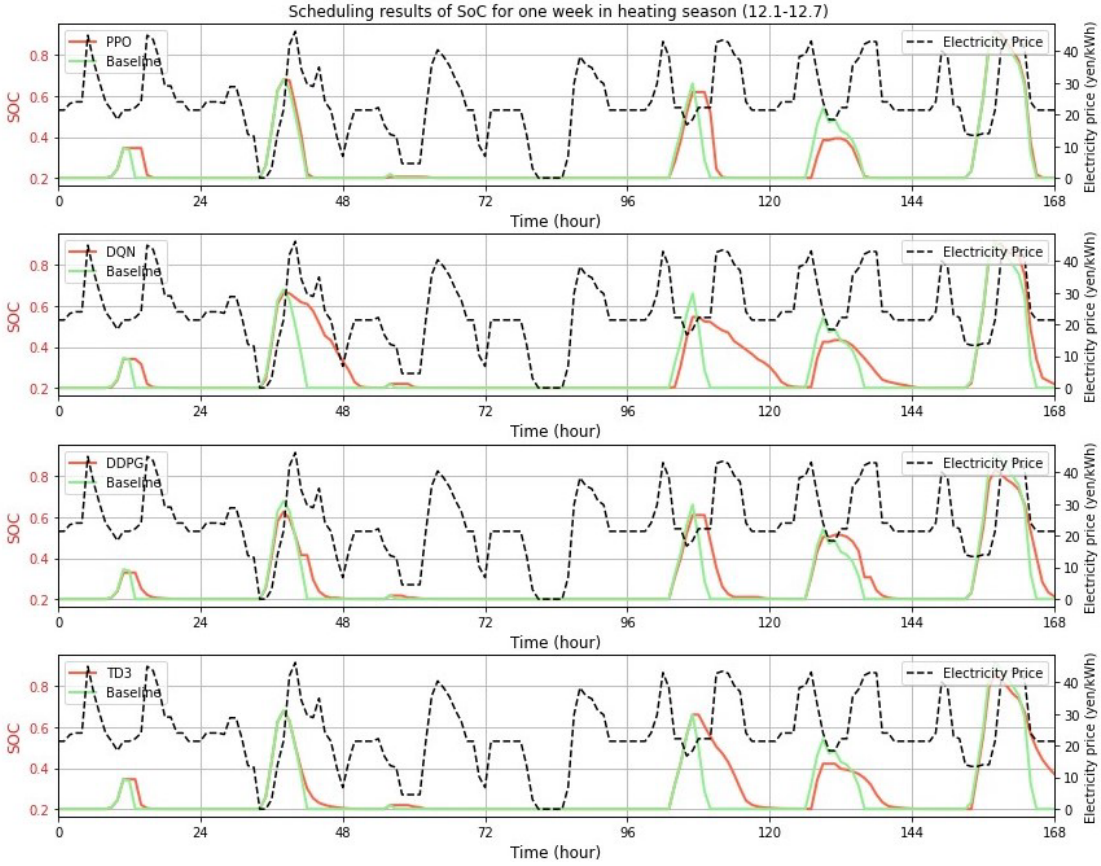

Fig. 6 One-week optimization results of the heating season

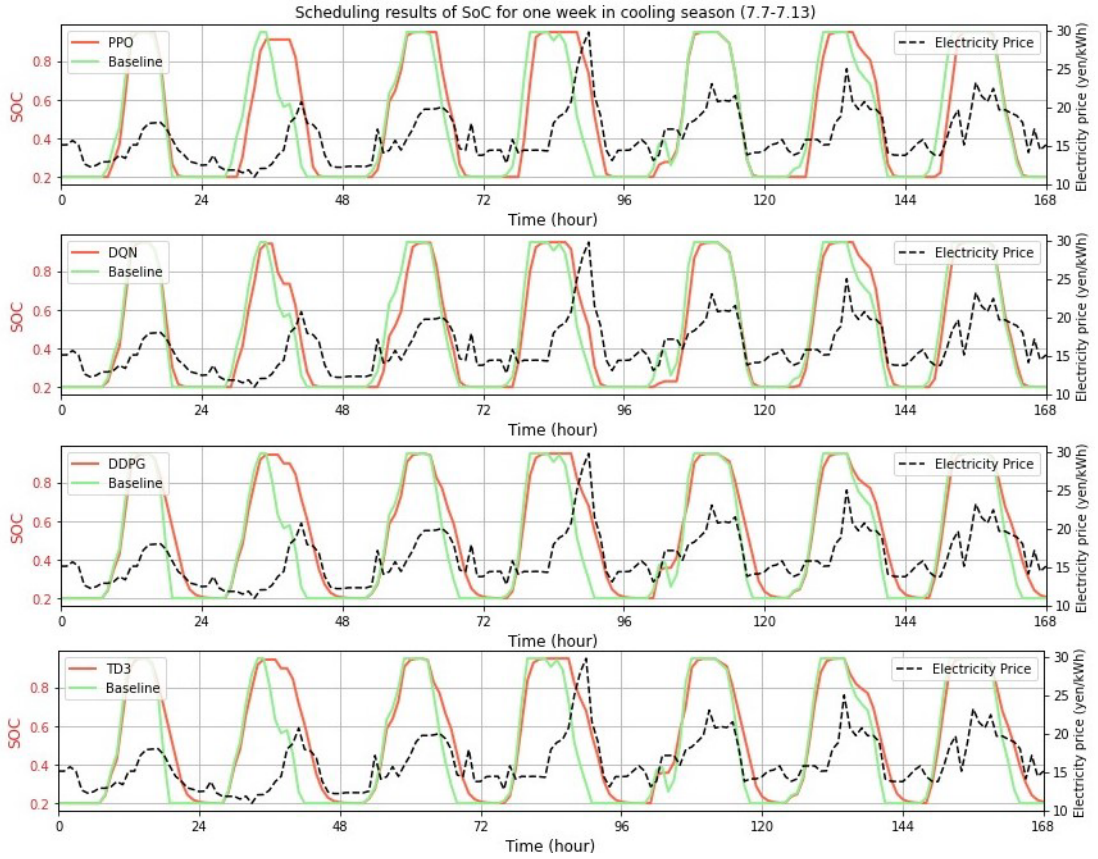

Fig. 7 One-week optimization results of the cooling season

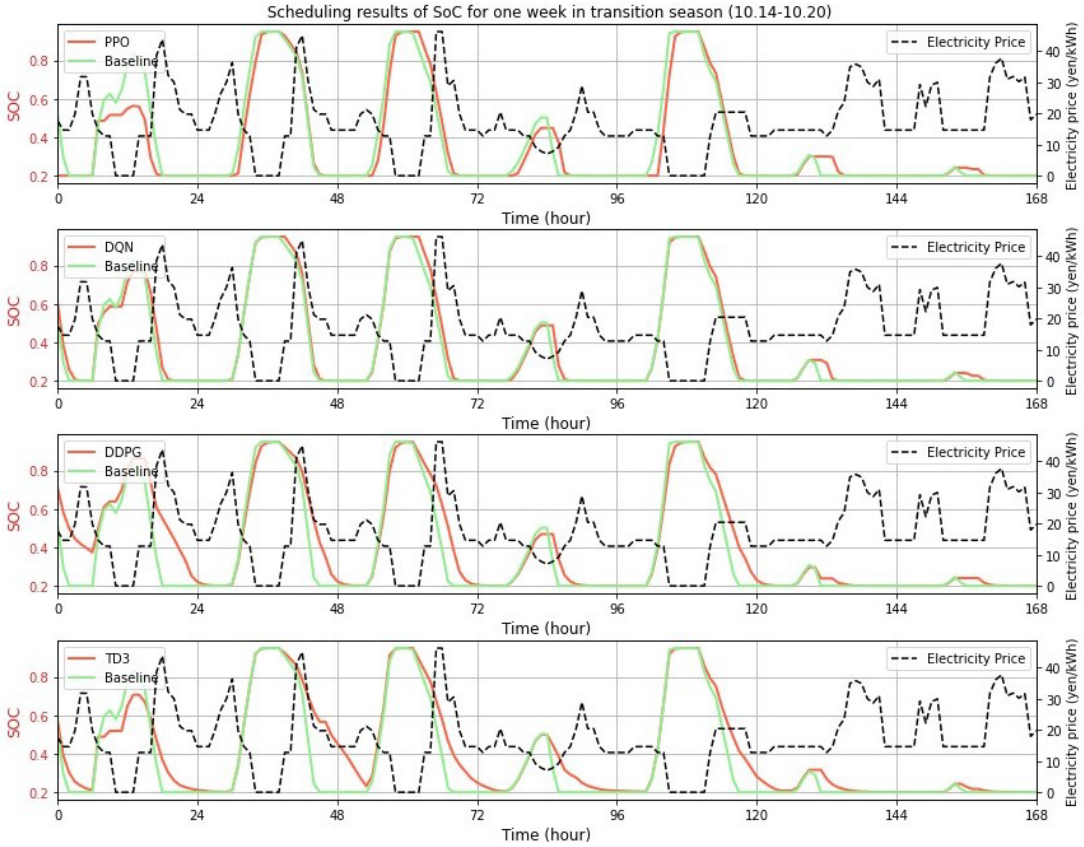

Fig. 8 One-week optimization results of the transition season

5. Conclusion

The RL-based energy control approach presents a promising potential for improving building energy efficiency due to their ability to learn strategies from environmental data and their scalability. This study proposed a model-based RL control method for operation optimization of the residential PV-battery system. The optimization goals aim at reducing the energy cost of the microgrid and ensuring that the PV self-consumption ratio is not lower than the baseline model. To achieve this goal, this study designed a new multi-objective optimization reward function, and experimental results proved the effectiveness of the designed reward function. We first set the benchmark strategy used by the target building as the baseline model. Then we adopted four advanced RL algorithms (PPO, DQN, DDPG, and TD3) to optimize the operation of the baseline model. The experimental results showed that the above four algorithms could achieve the optimization objective by using the designed reward function in this study. In addition, the TD3 algorithm had the best performance in each season of the year. It could reduce the annual energy costs by 17.82% and increase the PV self-consumption ratio by 0.86% compared with the baseline model.

In the case study, we analyzed the differences in optimization strategies between these four algorithms and evaluated their optimization efficiency during different periods. Although many RL-based energy management applications have been proposed, only a few studies have compared the optimization strategies between different RL algorithms in practical applications. This paper fills this gap, helping users better understand the performances of different algorithms in this scenario to facilitate the selection of RL algorithms for specific applications.

Future research will first focus on designing and optimizing reward functions in scenarios where additional energy sources (such as wind or fuel cells) and control objectives (such as heat pumps or air conditioners) are added[42–44]. Second, we will continue tuning these algorithms' hyperparameters to improve their generalization ability to deploy them in other buildings easily[45]. In addition, we will use the deep learning method to predict PV generation, demand, and electricity price and add this future information as observations to optimize the algorithm's performance.

Acknowledgments

This study was supported by Shandong Natural Science Foundation 'Research on Flexible District Integrated Energy System under High Penetration Level of Renewable Energy', grant number ZR2021QE084 and the Xiangjiang Plan 'Development of Smart Building Management

Technologies Towards Carbon Neutrality', grant number XJ20220028.

Reference

[1] Niu Z, Wu J, Liu X, Huang L, Nielsen P S. Understanding energy demand behaviors through spatio-temporal smart meter data analysis[J]. Energy, 2021, 226: 120493.

[2] Bogdanov D, Ram M, Aghahosseini A, Gulagi A, Oyewo A S, Child M, Caldera U, Sadovskaia K, Farfan J, De Souza Noel Simas Barbosa L, Fasihi M, Khalili S, Traber T, Breyer C. Low-cost renewable electricity as the key driver of the global energy transition towards sustainability[J]. Energy, 2021, 227: 120467.

[3] Li Y, Gao W, Ruan Y. Performance investigation of grid-connected residential PV-battery system focusing on enhancing self-consumption and peak shaving in Kyushu, Japan[J]. Renewable Energy, 2018, 127: 514–523.

[4] Komiyama R, Fujii Y. Assessment of post-Fukushima renewable energy policy in Japan's nation-wide power grid[J]. Energy Policy, 2017, 101: 594–611.

[5] Su Y, Zhou Y, Tan M. An interval optimization strategy of household multi-energy system considering tolerance degree and integrated demand response[J]. Applied Energy, 2020, 260: 114144.

[6] Li Y, Gao W, Zhang X, Ruan Y, Ushifusa Y, Hiroatsu F. Techno-economic performance analysis of zero energy house applications with home energy management system in Japan[J]. Energy and Buildings, 2020, 214: 109862.

[7] Zhao X, Gao W, Qian F, Ge J. Electricity cost comparison of dynamic pricing model based on load forecasting in home energy management system[J]. Energy, 2021, 229: 120538.

[8] Ullah Z, Elkadeem M R, Kotb K M, Taha I B M, Wang S. Multi-criteria decision-making model for optimal planning of on/off grid hybrid solar, wind, hydro, biomass clean electricity supply[J]. Renewable Energy, 2021, 179: 885–910.

[9] McIlwaine N, Foley A M, Morrow D J, Al Kez D, Zhang C, Lu X, Best R J. A state-of-the-art techno-economic review of distributed and embedded energy storage for energy systems[J]. Energy, 2021, 229: 120461.

[10] Al-Hinai A, Alyammahi H, Haes Alhelou H. Coordinated intelligent frequency control incorporating battery energy storage system, minimum variable contribution of demand response, and variable load damping coefficient in isolated power systems[J]. Energy Reports, 2021, 7: 8030–8041.

[11] Pallonetto F, De Rosa M, Finn D P. Impact of intelligent control algorithms on demand response flexibility and thermal comfort in a smart grid ready residential building[J]. Smart Energy, 2021, 2: 100017.

[12] Li Y, Xu W, Zhang X, Wang Z, Gao W, Xu Y. System value and utilization performance analysis of grid-integrated energy storage technologies in Japan[J]. Journal of Energy Storage, 2023, 63: 107051.

[13] Zhou Y, Ravey A, Péra M-C. Real-time cost-minimization power-allocating strategy via model predictive control for fuel cell hybrid electric vehicles[J]. Energy Conversion and

Management, 2021, 229: 113721.
[14] Stebel K, Fratczak M, Grelewicz P, Czeczot J, Klopot T. Adaptive predictive controller for energy-efficient batch heating process[J]. Applied Thermal Engineering, 2021, 192: 116954.
[15] Masero E, Maestre J M, Camacho E F. Market-based clustering of model predictive controllers for maximizing collected energy by parabolic-trough solar collector fields[J]. Applied Energy, 2022, 306: 117936.
[16] Koley S. Sustainability appraisal of arsenic mitigation policy innovations in West Bengal, India[J]. Infrastructure Asset Management, 2023, 10(1): 17–37.
[17] Monedero í, Barbancho J, Márquez R, Beltrán J F. Cyber-Physical System for Environmental Monitoring Based on Deep Learning[J]. Sensors, 2021, 21(11).
[18] Murray X, Apan A, Deo R, Maraseni T. Rapid assessment of mine rehabilitation areas with airborne LiDAR and deep learning: bauxite strip mining in Queensland, Australia[J]. Geocarto International, 2022, 37(26): 11223–11252.
[19] Ceusters G, Rodríguez R C, García A B, Franke R, Deconinck G, Helsen L, Nowé A, Messagie M, Camargo L R. Model-predictive control and reinforcement learning in multi-energy system case studies[J]. Applied Energy, 2021, 303: 117634.
[20] Zhang W, Wang J, Liu Y, Gao G, Liang S, Ma H. Reinforcement learning-based intelligent energy management architecture for hybrid construction machinery[J]. Applied Energy, 2020, 275: 115401.
[21] Wang Z, Hong T. Reinforcement learning for building controls: The opportunities and challenges[J]. Applied Energy, 2020, 269: 115036.
[22] Kolodziejczyk W, Zoltowska I, Cichosz P. Real-time energy purchase optimization for a storage-integrated photovoltaic system by deep reinforcement learning[J]. Control Engineering Practice, 2021, 106: 104598.
[23] Chen Y, Norford L K, Samuelson H W, Malkawi A. Optimal control of HVAC and window systems for natural ventilation through reinforcement learning[J]. Energy and Buildings, 2018, 169: 195–205.
[24] Haq E U, Lyu C, Xie P, Yan S, Ahmad F, Jia Y. Implementation of home energy management system based on reinforcement learning[J]. 2021 The 8th International Conference on Power and Energy Systems Engineering, 2022, 8: 560–566.
[25] Harrold D J B, Cao J, Fan Z. Data-driven battery operation for energy arbitrage using rainbow deep reinforcement learning[J]. Energy, 2022, 238: 121958.
[26] L. Yu, W. Xie, D. Xie, Y. Zou, D. Zhang, Z. Sun, L. Zhang, Y. Zhang, T. Jiang. Deep Reinforcement Learning for Smart Home Energy Management[J]. IEEE Internet of Things Journal, 2020, 7(4): 2751–2762.
[27] Y. Li, Y. Wen, D. Tao, K. Guan. Transforming Cooling Optimization for Green Data Center via Deep Reinforcement Learning[J]. IEEE Transactions on Cybernetics, 2020, 50(5): 2002–2013.
[28] E. Mocanu, D. C. Mocanu, P. H. Nguyen, A. Liotta, M. E. Webber, M. Gibescu, J. G. Slootweg. On-Line Building Energy Optimization Using Deep Reinforcement Learning[J]. IEEE Transactions on Smart Grid, 2019, 10(4): 3698–3708.
[29] Du Y, Zandi H, Kotevska O, Kurte K, Munk J, Amasyali K, Mckee E, Li F. Intelligent multi-zone residential HVAC control strategy based on deep reinforcement learning[J].

Applied Energy, 2021, 281: 116117.
[30] Lee H, Kim K, Kim N, Cha S W. Energy efficient speed planning of electric vehicles for car-following scenario using model-based reinforcement learning[J]. Applied Energy, 2022, 313: 118460.
[31] Sanayha M, Vateekul P. Model-based deep reinforcement learning for wind energy bidding[J]. International Journal of Electrical Power & Energy Systems, 2022, 136: 107625.
[32] Wang R. Reinforcement Learning: An Introduction[C]//2006 International Conference on Artificial Intelligence: 50 Years' Achievements, Future Directions and Social Impacts.
[33] Sutton R S, Precup D, Singh S. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning[J]. Artificial Intelligence, 1999, 112(1): 181–211.
[34] Sutton R, Barto A. Reinforcement Learning: An Introduction, Adaptive Computation and Machine Learning Series[J]. 1998. ,1998.
[35] Lehna M, Hoppmann B, Heinrich R, Scholz C. A Reinforcement Learning Approach for the Continuous Electricity Market of Germany: Trading from the Perspective of a Wind Park Operator[J]. 2021. ,2021.
[36] Lillicrap T P, Hunt J J, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D P. Continuous control with deep reinforcement learning[P]. 2020.
[37] Duryea E, Ganger M, Wei H. Deep Reinforcement Learning with Double Q-learning[J]. 2016. ,2016.
[38] Fujimoto S, Van Hoof H, Meger D. Addressing function approximation error in actor-critic methods[J]. Proceedings of the 35th International Conference on Machine Learning, Ser. Proceedings of Machine Learning Research, 2018, 80: 1587–1596.
[39] Grzes ´ M., Kudenko D. Plan-based reward shaping for reinforcement learning[Z]
[40] Brockman G, Cheung V, Pettersson L, Schneider J, Schulman J, Tang J, Zaremba W. OpenAI Gym[A]. arXiv,2016[2022-12-07].
[41] Raffin A, Hill A, Gleave A, Kanervisto A, Ernestus M, Dormann N. Stable-Baselines3: Reliable Reinforcement Learning Implementations[J]. Journal of Machine Learning Research, 2021, 22(268): 1–8.
[42] Gao Y, Matsunami Y, Miyata S, Akashi Y. Multi-agent reinforcement learning dealing with hybrid action spaces: A case study for off-grid oriented renewable building energy system[J]. Applied Energy, 2022, 326: 120021.
[43] Zhao L, Yang T, Li W, Zomaya A Y. Deep reinforcement learning-based joint load scheduling for household multi-energy system[J]. Applied Energy, 2022, 324: 119346.
[44] Harrold D J B, Cao J, Fan Z. Renewable energy integration and microgrid energy trading using multi-agent deep reinforcement learning[J]. Applied Energy, 2022, 318: 119151.
[45] Biemann M, Scheller F, Liu X, Huang L. Experimental evaluation of model-free reinforcement learning algorithms for continuous HVAC control[J]. Applied Energy, 2021, 298: 117164.