energies

MDPI

Article

Exploring the Preference for Discrete over Continuous Reinforcement Learning in Energy Storage Arbitrage

Jaeik Jeong *, Tai-Yeon Ku and Wan-Ki Park *

Energy ICT Research Section, Electronics and Telecommunications Research Institute, Daejeon 34129, Republic of Korea; kutai@etri.re.kr

* Correspondence: jaeik1210@etri.re.kr (J.J.); wkpark@etri.re.kr (W.-K.P.)

Abstract: In recent research addressing energy arbitrage with **energy storage systems (ESS**s), discrete reinforcement learning (RL) has often been employed, while the underlying reasons for this preference have not been explicitly clarified. This paper aims to elucidate why discrete RL tends to be more suitable than continuous RL for energy arbitrage problems. When using continuous RL, the charging and discharging actions determined by the agent often exceed the physical limits of the ESS, necessitating clipping to the boundary values. This introduces a critical issue where the learned actions become stuck at the state of charge (SoC) boundaries, hindering effective learning. Although recent advancements in constrained RL offer potential solutions, their application often results in overly conservative policies, preventing the full utilization of ESS capabilities. In contrast, discrete RL, while lacking in granular control, successfully avoids these two key challenges, as demonstrated by simulation results showing superior performance. Additionally, it was found that, due to its characteristics, discrete RL more easily drives the ESS towards fully charged or fully discharged states, thereby increasing the utilization of the storage system. Our findings provide a solid justification for the prevalent use of discrete RL in recent studies involving energy arbitrage with ESSs, offering new insights into the strategic selection of RL methods in this domain. Looking ahead, improving performance will require further advancements in continuous RL methods. This study provides valuable direction for future research in continuous RL, highlighting the challenges and potential strategies to overcome them to fully exploit ESS capabilities.

Keywords: energy arbitrage; energy storage system; discrete reinforcement learning; continuous reinforcement learning; constrained reinforcement learning

Citation: Jeong, J.; Ku, T.-Y.; Park, W.-K. Exploring the Preference for Discrete over Continuous Reinforcement Learning in Energy Storage Arbitrage. *Energies* **2024**, *17*, 5876. https://doi.org/10.3390/en17235876

Academic Editors: Riccardo Berta, Matteo Nardello and Luca Lazzaroni

Received: 26 October 2024
Revised: 15 November 2024
Accepted: 21 November 2024
Published: 22 November 2024

1. Introduction

Energy arbitrage is a key approach for optimizing energy storage system (ESS) usage, especially with the rise of renewable energy sources [1]. By buying electricity when prices are low and selling it when prices are high, energy arbitrage balances supply and demand, stabilizes the grid, and maximizes profits. Various storage technologies, such as battery storage, hydrogen storage, and thermal storage, can be used for energy arbitrage [2,3]. Battery storage is particularly effective due to its rapid response to real-time price changes, making it well-suited for dynamic energy markets [4,5]. This paper focuses on battery storage for its suitability in real-time energy arbitrage and its significant potential for profit maximization.

Real-time energy markets are increasingly characterized by high variability and dynamic pricing, largely due to the penetration of renewable energy sources, whose supply fluctuates based on environmental conditions. This variability demands that ESSs be able to respond quickly and efficiently to market signals in order to maximize the economic benefits of energy arbitrage. As a result, batteries are well-suited for energy arbitrage in environments with high real-time price volatility, because they can ramp up or down power

output within seconds [6]. Traditional optimization-based methods, such as linear programming and rule-based control, have been used to solve the energy arbitrage problem [7]. However, these methods struggle with the stochastic and dynamic nature of renewable energy and real-time pricing. They require detailed models that can be challenging to develop and often fail to adapt to unexpected changes. Reinforcement learning (RL), by contrast, can learn optimal control strategies through interaction with the environment, making it well-suited for complex, unpredictable conditions [4,5]. RL's adaptability to changing market conditions is essential for effective management of ESSs.

Within the RL framework, there are two main approaches for action selection: discrete RL and continuous RL [8]. Discrete RL involves selecting from a finite set of predefined actions, such as charging or discharging at specific rates. This approach simplifies the learning process by reducing the complexity of the action space, making it easier to learn effective policies, especially in environments with high variability. This simplicity makes discrete RL more efficient in adhering to system constraints while still optimizing performance. Discrete actions can lead to faster convergence because the agent has fewer options to evaluate, which helps in stabilizing the learning process. Conversely, continuous RL allows for actions that can take any value within a specified range, providing theoretically finer control over the charging and discharging processes. This allows the agent to optimize the systems with greater precision, which could potentially result in higher profits. In energy arbitrage, actions often involve determining the amount of charging or discharging, which is inherently a continuous value. Thus, continuous RL is considered the common choice, and several studies have addressed the energy arbitrage problem using continuous RL [9,10]. However, discrete RL, which selects from predefined actions such as fixed charging or discharging rates, is often preferred due to its simplicity and stability [4,11–13].

Our study aims to clarify why discrete RL often outperforms continuous RL in energy arbitrage scenarios. The primary objective of this research is to experimentally demonstrate that discrete RL is a better choice for energy arbitrage involving ESSs, not only in terms of simplicity and stability but also in terms of performance. Previous research has suggested that discrete RL is preferred mainly because it is simple and stable, but does not clearly mention the problems that could occur when using continuous RL [4,11–13]. This paper goes further by demonstrating that, even when considering performance, discrete RL remains a superior choice for energy arbitrage. We aim to provide clear evidence that discrete RL's advantages are not just about ease of implementation but also about achieving better practical results in real-world energy markets. While it has been suggested that continuous RL might yield better performance due to its finer control capabilities, our findings demonstrate otherwise. One major issue with continuous RL is action clipping, which keeps the State of Charge (SoC) within acceptable bounds. This often leads to the SoC becoming stuck in fully charged or discharged states, hindering exploration across the full range of SoC values.

To prevent the agent from becoming stuck, constrained RL is often used [14]. Constrained RL incorporates specific constraints that must be adhered to at every time step while maximizing cumulative rewards. These constraints are typically reformulated using a Lagrangian approach, where a Lagrangian multiplier is introduced to impose penalties when constraints are violated [15]. In energy arbitrage, the constraint is set to ensure that the SoC does not exceed specific bounds, with penalties applied proportionally to any violations of these bounds. The key challenge here is appropriately setting the value of the Lagrangian multiplier. If set too conservatively, the agent's learning may focus solely on satisfying constraints, thereby resulting in overly cautious behavior that restricts the full utilization of ESS capabilities. In energy arbitrage problems, discrete RL can select the best action among those that satisfy the constraints, without requiring action clipping or Lagrangian multiplier tuning. This simplicity makes discrete RL more efficient in adhering to system constraints while still optimizing performance. Most studies on ESS control based on continuous control have successfully used constrained RL to ensure that the SoC remains within a specified range, thereby enhancing the stability of the learning process [16–18].

However, to the best of our knowledge, few studies in the energy arbitrage domain have specifically evaluated the effectiveness of constrained RL. Thus, it is necessary to investigate and validate the impact of constrained RL in the context of energy arbitrage as well.

In our research, we applied a continuous RL algorithm for energy arbitrage using real energy price data and tuned the Lagrangian multiplier to find the optimal setting. Despite this setting, continuous RL still underperformed compared with discrete RL. On the other hand, discrete RL naturally guided the system to fully charged or discharged states, thus maximizing ESS utilization. Continuous RL's conservative approach in satisfying constraints led to less efficient outcomes, proving that discrete RL provides a more practical and effective solution for energy arbitrage in its current state. Specifically, our results showed that discrete RL outperformed continuous RL by 42% in terms of overall performance, demonstrating its clear advantage in maximizing the efficiency of energy arbitrage.

Our findings highlight the need for a nuanced approach to energy arbitrage research. The key contribution of this study is to demonstrate that, when considering continuous control for improving energy arbitrage performance, simply replacing discrete RL with continuous RL is not sufficient and can even degrade performance. Instead, it is crucial to develop continuous RL algorithms that incorporate the strengths of discrete RL. By providing experimental evidence of the limitations of continuous RL in energy arbitrage, this research paves the way for more informed decision-making in selecting RL methodologies and underscores the importance of addressing specific limitations for future advancements. Future research should focus on modifying continuous RL algorithms to mitigate overly conservative behavior, such as adapting penalty functions dynamically or employing hybrid approaches that combine the flexibility of continuous actions with the robustness of discrete action selection. These modifications could help in fully exploiting the potential of ESSs while avoiding the pitfalls observed in purely continuous RL strategies.

The rest of this paper is organized as follows. In Section 2, we discuss related works, focusing on traditional optimization-based energy arbitrage, discrete RL-based energy arbitrage, and continuous RL-based energy arbitrage. In Section 3, we describe the methodologies for the discrete RL and continuous RL approaches for energy arbitrage. In Section 4, energy arbitrage is conducted on real-world energy price data for performance evaluations, followed by the conclusion in Section 5.

2. Related Works

In this section, we provide an overview of the existing research related to energy arbitrage. We categorize related works into three primary approaches.

2.1. Traditional Optimization-Based Energy Arbitrage

Traditional optimization techniques, such as linear programming [7], mixed-integer linear programming [19], and dynamic programming [20], have been widely used to solve energy arbitrage problems by optimizing the charging and discharging schedules of ESSs. These methods provide optimal solutions in well-defined environments, but struggle to adapt to the high variability and uncertainty of real-time energy markets. Although effective under deterministic conditions, they require detailed system models, which can be computationally expensive and may not respond well to sudden market changes. Despite these limitations, traditional optimization approaches have laid a solid foundation for energy arbitrage and are still used as benchmark methods in the field. Furthermore, using stochastic dynamic programming, a form of stochastic optimization, can partially address the uncertainty challenges when solving the energy arbitrage problem [21]. However, with the growing complexity and uncertainty in energy systems due to increased distributed energy resources, traditional optimization methods face significant challenges.

2.2. Discrete RL-Based Energy Arbitrage

Recent research has increasingly focused on reinforcement learning (RL) to address the dynamic nature of energy arbitrage, with discrete RL approaches gaining significant

attention. Discrete RL involves selecting from a predefined set of charging or discharging actions, making it particularly effective for environments with high variability. The reduced complexity of the action space simplifies the learning process, often leading to faster convergence and more stable policies compared with continuous approaches. This also allows for better adherence to system constraints without the need for action clipping or Lagrangian multiplier tuning. Studies such as those by [4,11–13] have demonstrated the robustness and efficiency of discrete RL in energy arbitrage applications. These studies have shown that discrete RL can effectively balance computational simplicity with operational performance, making it an attractive solution for ESS management under uncertain and fluctuating market conditions. However, many discrete RL-based energy arbitrage studies have not explicitly explained why discrete RL was chosen over continuous RL. The work in [11] suggested that continuous RL could improve future performance, but a deeper examination of the specific challenges that arise when using continuous RL instead of discrete RL is still needed. The work in [12] argued that discrete RL could achieve a performance comparable to continuous RL due to the emergence of bang-bang behavior in continuous RL [22], but the comparison and analysis between the two algorithms were not provided. Additionally, [13] argued that discrete RL outperformed continuous RL, attributing this to instability and the difficulty of hyperparameter tuning. However, no comparison was made after addressing the instability using constrained RL.

2.3. Continuous RL-Based Energy Arbitrage

Continuous RL has also been explored as an alternative approach to provide finer control over charging and discharging actions. By allowing continuous-valued actions, these methods theoretically offer greater precision in optimizing ESS schedules. This advantage has made continuous RL widely used for optimizing ESS schedules in general. In most cases, continuous RL employs constrained RL to ensure that the SoC remains within specified bounds. Many ESS control studies have combined continuous RL with constrained RL to solve problems such as home energy management [16], dispatch planning [17], and fast charging [18]. However, in the energy arbitrage problem, the difference between using discrete RL and continuous RL has not been thoroughly compared and analyzed. In [5], continuous RL has shown potential for improving energy arbitrage indirectly, but direct effects have been limited. The works in [9,10] addressed energy arbitrage directly using continuous RL, but no comparison was made with discrete RL. Therefore, there is a need to analyze the effectiveness of continuous RL in energy arbitrage specifically and compare its performance directly with discrete RL.

3. Methods

In this section, we present the methodologies used to approach the energy arbitrage problem using both discrete and continuous RL strategies. We first describe the energy arbitrage problem and the battery model employed, followed by the specific implementations of the discrete RL and continuous RL methods. Although this paper focuses on batteries, other types of ESS, such as thermal storage systems, can be effectively modeled as well [23].

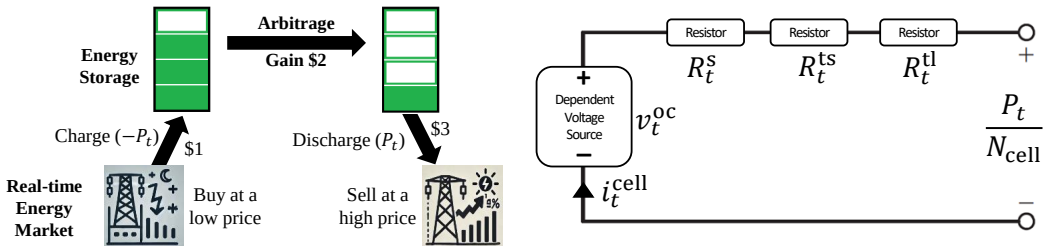
3.1. Energy Arbitrage and Battery Model

The overall system model shown in Figure 1a mainly consists of the battery and the real-time energy market. The model to calculate the profit at time t can be expressed as

$$r_t = c_t \cdot P_t \cdot \Delta t, \tag{1}$$

where r_t represents the revenue at time t, c_t is the energy price at time t, P_t is the charge or discharge volume at time t (negative for charging, positive for discharging), and Δt denotes the duration of the time slot. Since c_t is not known in advance, it must be predicted, which introduces a forecasting component into the model. In this model, r_t is later used as the reward signal in a reinforcement learning framework, where the goal is to optimize ESS

operations to maximize profit by strategically charging and discharging based on price fluctuations.



(**a**) (**b**)

Figure 1. System models.(**a**) Energy storage arbitrage process. (**b**) Steady-state battery cell equivalent circuit.

Since there are the charging and discharging efficiencies, denoted by η_t^c and η_t^d, respectively, all charging power cannot be stored in the battery, and all discharging power cannot be sold to the real-time market. To calculate the η_t^c and η_t^d, a steady-state equivalent electrical circuit of the lithium ion battery cell shown in Figure 1b can be used [24]. The circuit consists of an open circuit voltage, v_t^{oc}, and series resistors, R_t^{s}, R_t^{ts}, and R_t^{tl}, which represent ohmic losses, charge transfer, and membrane diffusion, respectively. R_t^{ts} and R_t^{tl} are each connected in parallel with a capacitor, but for simplicity SoC is treated as constant within one time slot, which makes direct current during the time duration and enables ignoring of the capacitors [25].

The open circuit voltage, v_t^{oc}, and the three resistors, R_t^{s}, R_t^{ts}, and R_t^{tl}, are determined by the battery SoC at time slot t, denoted by SoC_t, where the relations are described as

$$v_t^{\text{oc}} = \mathfrak{a}_0 e^{-\mathfrak{a}_1 \text{SoC}_t} + \mathfrak{a}_2 + \mathfrak{a}_3 \text{SoC}_t - \mathfrak{a}_4 (\text{SoC}_t)^2 + \mathfrak{a}_5 (\text{SoC}_t)^3, \tag{2a}$$

$$R_t^{\text{s}} = \mathfrak{b}_0 e^{-\mathfrak{b}_1 \text{SoC}_t} + \mathfrak{b}_2 + \mathfrak{b}_3 \text{SoC}_t - \mathfrak{b}_4 (\text{SoC}_t)^2 + \mathfrak{b}_5 (\text{SoC}_t)^3, \tag{2b}$$

$$R_t^{\text{ts}} = \mathfrak{c}_0 e^{-\mathfrak{c}_1 \text{SoC}_t} + \mathfrak{c}_2, \tag{2c}$$

$$R_t^{\text{tl}} = \mathfrak{e}_0 e^{-\mathfrak{e}_1 \text{SoC}_t} + \mathfrak{e}_2, \tag{2d}$$

$$R_t = R_t^{\text{s}} + R_t^{\text{ts}} + R_t^{\text{tl}}, \tag{2e}$$

where all Fraktur typefaces in the equations are constant battery cell parameters. Then, i_t^{cell} can be obtained by solving the following quadratic equations,

$$P_t = \begin{cases} N_{\text{cell}} \cdot \left(v_t^{\text{oc}} \cdot i_t^{\text{cell}} + \left(i_t^{\text{cell}} \right)^2 \cdot R_t \right), & \text{if } P_t < 0 \text{ (charging)}, \\ N_{\text{cell}} \cdot \left(v_t^{\text{oc}} \cdot i_t^{\text{cell}} - \left(i_t^{\text{cell}} \right)^2 \cdot R_t \right), & \text{if } P_t \geq 0 \text{ (discharging)}, \end{cases} \tag{3}$$

where N_{cell} is the number of battery cells.

The charging efficiency, η_t^c, is determined by the ratio of the absorbing power of the voltage source and the charging power. Likewise, the discharging efficiency, η_t^d, is determined by the ratio of the discharging power and the supplying power of the voltage source. Then, η_t^c and η_t^d are given by

$$\eta_t^c = \frac{v_t^{\text{oc}} \cdot i_t^{\text{cell}} \cdot N_{\text{cell}}}{P_t}, \tag{4a}$$

$$\eta_t^d = \frac{P_t}{v_t^{\text{oc}} \cdot i_t^{\text{cell}} \cdot N_{\text{cell}}}. \tag{4b}$$

Accordingly, the SoC evolves in time as follows:

$$\mathrm{SoC}_{t+1} = \begin{cases} \mathrm{SoC}_t - \eta_t^c \frac{P_t}{E_{\max}} \Delta t, & \text{if } P_t < 0 \text{ (charging)}, \\ \mathrm{SoC}_t - \frac{1}{\eta_t^d} \frac{P_t}{E_{\max}} \Delta t, & \text{if } P_t \geq 0 \text{ (discharging)}, \end{cases} \tag{5}$$

where $E_{\max}$ is a total battery capacity. In general, the efficiency of battery lies in $[0.96, 0.995]$ and becomes high for high SoC and low charging/discharging power. When determining P_t, the charging and discharging power limitations, denoted by $P_t^{\min}$ and $P_t^{\max}$, should be examined first, and are determined by the SoC_t. Battery degradation is known to be severe at both ends of the SoC, which implies that SoC_t should be constrained as $\mathrm{SoC}_{\min} \leq \mathrm{SoC}_t \leq \mathrm{SoC}_{\max}$. Then, $P_t^{\min}$ and $P_t^{\max}$ are determined by the following equations:

$$\mathrm{SoC}_{\max} = \mathrm{SoC}_t - \eta_t^c \frac{P_t}{E_{\max}} \Delta t, \quad \text{when } P_t = P_t^{\min} \text{ (charging limitation)}, \tag{6a}$$

$$\mathrm{SoC}_{\min} = \mathrm{SoC}_t - \frac{1}{\eta_t^d} \frac{P_t}{E_{\max}} \Delta t, \quad \text{when } P_t = P_t^{\max} \text{ (discharging limitation)}. \tag{6b}$$

The current SoC is naturally taken into account when determining the charging or discharging values, P_t. However, as shown in Equation (5), the current decision also affects the SoC of the subsequent time slot. Additionally, since real-time energy prices, c_t, are not known in advance, they must be predicted. As a result, determining the value of P_t becomes a sequential decision-making problem under uncertainty, which justifies the use of reinforcement learning (RL). Since the value of P_t depends on both c_t and SoC_t, these two variables form the state in the RL framework for this problem. While the predicted value of c_t can be used directly [4], it is also possible to introduce the concept of a partially observable state, referred to as an observation [5]. Thus, the observation, o_t, and state, s_t, are defined as follows.

$$o_t = (c_{t-1}, \mathrm{SoC}_t), \tag{7a}$$

$$s_t = (o_0, o_1, \cdots, o_{t-1}, o_t). \tag{7b}$$

Given that s_t is composed of a sequential series of observations, long short-term memory (LSTM) can be utilized as the RL model to capture the temporal dependencies in the decision-making process [5]. The action, a_t, can be defined as the charging or discharging amount, P_t. However, as shown in Equation (6), P_t is bounded by a minimum and maximum value, and the action output by the RL model may exceed these limits. To prevent this, clipping is applied to ensure that the action stays within the allowable range as follows:

$$P_t = \mathrm{clip}(a_t, P_t^{\min}, P_t^{\max}). \tag{8}$$

The reward, r_t, in the RL framework corresponds to the revenue or cost from energy arbitrage in the real-time market, as defined in Equation (1).

3.2. Discrete RL Method: Deep Q-Network (DQN)

We first explore discrete RL. One of the most widely used algorithms in discrete RL is the deep Q-network (DQN) [8]. DQN combines the traditional Q-learning algorithm with deep neural networks, allowing it to efficiently learn in environments with large state spaces. DQN uses a neural network to approximate Q-values, outputting the Q-values for all possible actions given the current state. The action with the highest Q-value is then selected. Additionally, DQN introduces two key techniques—experience replay and the target network—to improve the stability of learning. Let $Q_\omega(s_t, a_t)$ be the action-value function when taking action a_t in state s_t, where the parameter ω represents the weights of the neural network used to approximate the Q-function. The loss function, $\mathcal{L}_Q(\omega)$, is then defined as the mean squared error (MSE) between the predicted Q-value and the target Q-value, which is computed as:

$$\mathcal{L}_Q(\omega) = \mathbb{E}_{(s_t,a_t,r_t,s_{t+1})\sim D}\left[\left(r_t + \gamma \max_a Q_{\omega^-}(s_{t+1},a) - Q_\omega(s_t,a_t)\right)^2\right] \quad (9)$$

where γ is the discount factor that determines the importance of future rewards, D is the replay buffer, and ω^- is the parameters of the target network.

To apply DQN to the energy arbitrage problem, it is necessary to discretize the action space, as illustrated in Figure 2a. If the action space is discretized too finely, the complexity of the problem increases, which can negatively impact performance. In this paper, we follow the approach used in previous studies by discretizing the actions into three categories: fully charged, idle, and fully discharged [11,12]. This simplification is motivated by the typical strategy of charging as much as possible when energy prices are low and discharging as much as possible when prices are high. In this case, the action, a_t, can take one of three values: $P_t^{\min}$, 0, or $P_t^{\max}$, corresponding to fully charging, idling, or fully discharging, respectively. As a result, there is no need for action clipping, as described in Equation (8), since a_t is inherently constrained to these three discrete values. This naturally resolves the issue of limiting the charge and discharge volumes, simplifying the implementation and ensuring that the actions remain within the allowable operational limits.

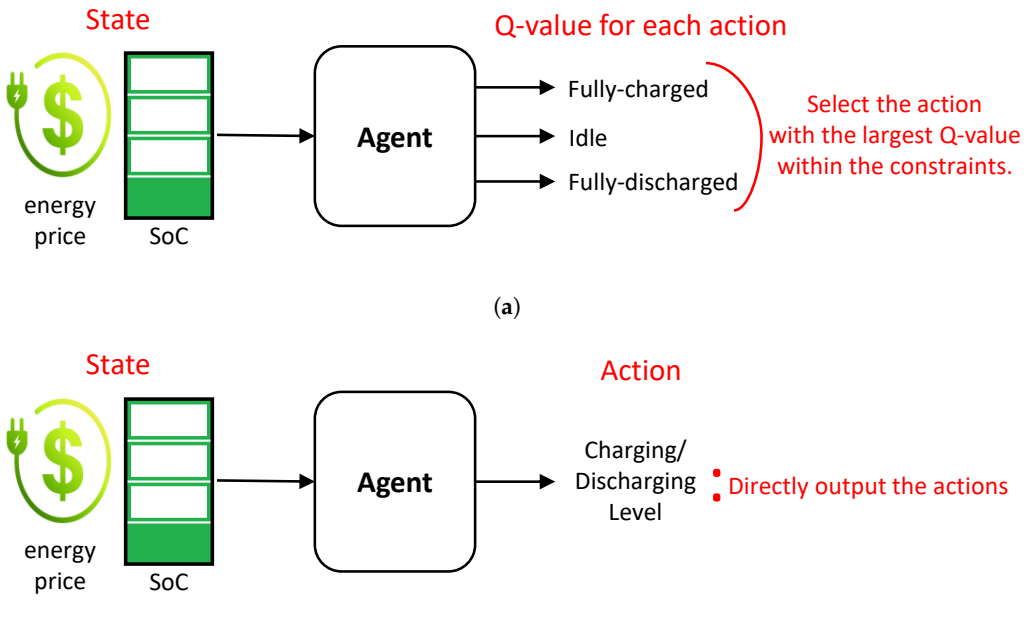


Figure 2. Illustration of the RL mechanisms for ESS control. (**a**) Discrete RL mechanism for ESS control. (**b**) Continuous RL mechanism for ESS control.

3.3. Continuous RL Method: Advantage Actor–Critic (A2C)

Next, we explore continuous RL. In the energy arbitrage problem, using discrete RL requires discretizing the action space, specifically the charge and discharge amounts. Since these quantities are naturally continuous, it is worth investigating the use of continuous RL. As shown in Figure 2b, continuous RL simplifies the action output by reducing it to a single value, which directly represents the action itself. When using discrete RL, increasing the granularity of action discretization results in an excessive number of discrete actions, leading to a larger set of Q-values to compare, which can reduce efficiency. In contrast, continuous RL processes the action as a single output value, making it more suitable for fine-grained control and potentially more efficient in environments where continuous action spaces are required.

A key consideration when applying continuous RL is that the actions output by the RL model are not inherently constrained within a specific range. In theory, the action values could range from negative infinity to positive infinity. However, in our model, the action values are restricted to lie between $P_t^{\min}$ and $P_t^{\max}$. One might assume that this issue is

resolved by applying action clipping, as described in Equation (8), which automatically clips the action values within the allowable range of $P_t^{\min}$ and $P_t^{\max}$. However, simply relying on clipping can introduce instability during training. This problem is illustrated in Figure 3, where an example demonstrates the issue. If the maximum discharge rate is limited to 75%, and the model outputs an action that suggests an 80% discharge, clipping would reduce this to 75%. From the agent's perspective, this creates a situation where changing the action value from 80% results in no change in the reward, leading the agent to perceive no benefit from adjusting the action in this range. This can ultimately halt learning, as the agent fails to recognize the differences in rewards for actions near the boundaries—a phenomenon that can be viewed as a form of vanishing gradient. While increasing the exploration rate might help the agent detect changes in the reward for different actions, this comes at the cost of reduced exploitation, hindering the agent's ability to fully optimize its policy.

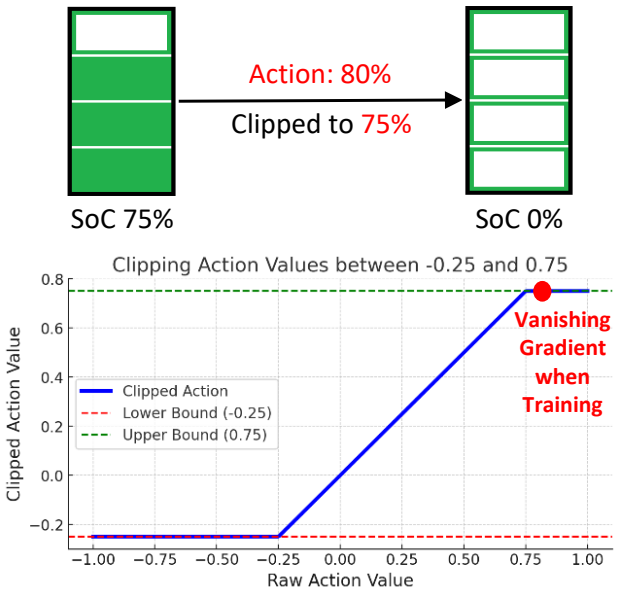


Figure 3. Example of the continuous RL with clipped action.

Due to this issue, constrained RL is often employed when controlling ESSs with continuous RL [16–18]. In constrained RL, an additional condition is imposed to ensure that the action values remain within the range of $P_t^{\min}$ and $P_t^{\max}$ for every time step t [14]. This means that the agent must not only learn to maximize the cumulative reward but also satisfy the imposed constraints. Given that the constraint in this study was limited to maintaining the SoC within a defined range, we selected the Lagrangian multiplier method due to its simplicity and effectiveness in handling this specific requirement [15]. The Lagrangian multiplier effectively penalizes the agent whenever the action exceeds the allowable bounds, providing a penalty to the reward proportionate to the extent to which the constraints are violated. When applying constrained RL to our model, the reward function is modified accordingly. In addition to maximizing the original reward from energy arbitrage, the agent receives a penalty when the action value exceeds the $P_t^{\min}$ or $P_t^{\max}$. This adjustment to the reward structure ensures that the agent learns to operate within the allowable charge and discharge rates while optimizing its long-term performance in terms of cumulative reward. The modified reward function in our model, denoted by r_t^c, is expressed as follows.

$$r_t^c = c_t \cdot P_t \cdot \Delta t - \lambda |a_t - P_t|. \tag{10}$$

As can be seen in the equation, the agent is penalized by the difference between a_t and P_t. This is because, when P_t remains within the allowable range of $P_t^{\min}$ and $P_t^{\max}$, the values of a_t and P_t are equal.

Figure 4 illustrates an example related to this mechanism. When the maximum discharge rate is limited to 75%, and the agent selects an action corresponding to an 80% discharge, a penalty proportional to the 5% excess is applied to the reward. This penalty discourages the agent from choosing actions that exceed the operational limits. As a result, the agent is able to recognize that adjusting its action beyond the 75% threshold leads to a decrease in reward due to the penalty, thus avoiding the situation where no difference in reward is perceived when changing the action around the boundary. Since the penalty is proportional to the amount by which the action violates the constraint, the agent can effectively learn to stay within the prescribed bounds while still optimizing for long-term cumulative rewards. This proportional penalty approach ensures that the agent not only seeks to maximize performance but also respects the operational constraints, preventing the learning stagnation that can occur when constraints are ignored. A critical point in this approach is the proper tuning of the Lagrangian multiplier, λ [26]. If λ is too large, the agent may focus excessively on satisfying the constraints at the expense of maximizing the cumulative reward. Conversely, if λ is too small, the constraints may become insignificant, allowing the agent to frequently violate them. Therefore, it is essential to find an appropriate balance for λ to ensure the agent optimizes its policy while adhering to the system's operational limits.

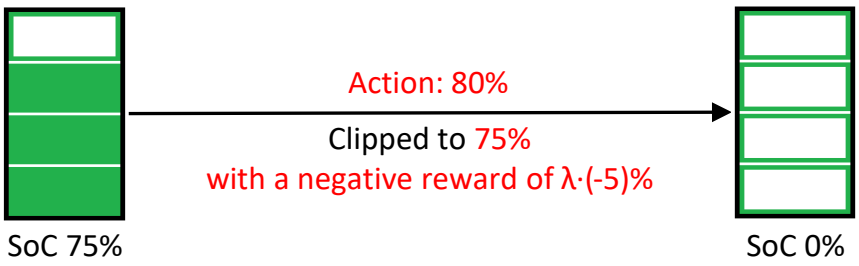


Figure 4. Example of the continuous RL with clipped action with Lagrangian multiplier.

Now, we explore continuous RL, where the most fundamental algorithm is advantage actor–critic (A2C) [8]. A2C is a policy gradient-based method that separates the decision-making process into two components: the actor and the critic. The actor is responsible for selecting actions based on the current policy, while the critic evaluates the performance of the actor's actions by estimating the value function. A2C introduces the concept of advantage, which measures how much better an action is compared with the average action taken from a given state. The objective of A2C is to maximize the advantage of the actions chosen by the actor. The critic helps stabilize the training by providing feedback to the actor, ensuring that the agent learns more efficiently in environments with continuous action spaces. This separation of roles allows A2C to handle more complex environments and policies compared with traditional value-based methods. Let $\pi_\theta(a_t|s_t)$ be the probability density of taking action a_t in state s_t with parameter θ, which is learned by the actor. Also, let $V_\theta(s_t)$ be the state-value function in state s_t approximated by parameter θ, which is learned by the critic. Note that parameter θ is shared between the actor and the critic since the inputs include only the state. The actor loss function, $J_\pi(\theta)$, and the critic loss function, $\mathcal{L}_V(\theta)$, are then defined as follows:

$$\nabla_\theta J_\pi(\theta) \triangleq \hat{\mathbb{E}}_t\left[\frac{\nabla_\theta \pi_\theta(a_t|s_t)}{\pi_\theta(a_t|s_t)}\left(r_t^c + \gamma V_\theta(s_{t+1}) - V_\theta(s_t)\right)\right], \tag{11}$$

$$\mathcal{L}_V(\theta) = \hat{\mathbb{E}}_t\left[\left(r_t^c + \gamma V_\theta(s_{t+1}) - V_\theta(s_t)\right)^2\right], \tag{12}$$

where the expectation $\hat{\mathbb{E}}_t[\cdot]$ indicates the empirical average over a finite batch of samples.

A2C cannot utilize replay buffer because it is an on-policy algorithm, meaning that it can only use the most recent experiences generated by the current policy for learning. This limitation requires alternative methods to ensure the stability of training. One of the simplest approaches to enhance learning stability in A2C is to use generalized advantage estimation instead of the standard advantage estimation formula. It smooths out the variance in the advantage function, leading to more stable and reliable updates. Additionally, to further stabilize the learning process, the policy update can be clipped within a range of $1 \pm \epsilon$, preventing excessively large policy changes during training. This clipping mechanism ensures that the updates are constrained, reducing the likelihood of destabilizing the learning process. The algorithm that incorporates these methods into A2C is known as proximal policy optimization (PPO), which has become widely used for its ability to maintain stable performance while optimizing policies in continuous action spaces [27]. We confirmed that the clipping mechanism improves training stability in our environment. Consequently, PPO was selected for its enhanced stability over standard A2C. However, in cases where clipping offers no added benefit, the lower computational complexity of standard A2C can make it a more efficient alternative.

Unlike DQN, which is applicable only to discrete RL, both A2C and PPO can be applied to both discrete RL and continuous RL problems. This flexibility makes them more versatile in handling a wider range of RL tasks. However, when the problem is limited to discrete action spaces, DQN is often preferred for its simplicity and efficiency. DQN's value-based approach can offer faster convergence and lower computational complexity compared with policy-gradient methods like A2C or PPO, making it a suitable choice for problems where speed and simplicity are priorities [28]. For instance, when applying RL to tiny machine learning (TinyML) environments, DQN is often a better choice than A2C or PPO. TinyML involves deploying models on resource-constrained devices with limited computational power and memory. Since DQN is a simpler and more lightweight algorithm compared with the more computationally intensive A2C or PPO, it is better suited for such environments. DQN's lower computational requirements make it more suitable for TinyML environments, such as real-time applications, compared with the higher complexity of A2C or PPO. Studies in various fields show that models with lower complexity, like the approach used in our study, can achieve fast and accurate results. For example, Ref. [29] demonstrates that traditional optimization techniques can yield results comparable to neural networks while maintaining faster computation times. Similarly, Ref. [30] shows how traditional machine learning methods can enhance deep learning performance when integrated, emphasizing the importance of model simplicity in real-world applications.

4. Performance Evaluation

4.1. Experimental Setup

In this section, we evaluate the performance of both discrete RL and continuous RL methods. For the continuous RL approach, we experimented with three different values of the Lagrangian multiplier, λ: a very small value ($\lambda = 0$), a very large value ($\lambda = 1$), and the most appropriate value ($\lambda = 0.1$). The value of $\lambda = 0.1$ was selected based on empirical results showing the best performance during testing. We demonstrated the effectiveness of the proposed methods through energy arbitrage experiments, using real energy price data from the 2017 U.K. wholesale market [31]. Specifically, we used the first 2000 data points, sampled every 30 min, and divided the dataset into a training set (1000 data points), a validation set (500 data points), and a test set (500 data points) in chronological order. The validation set was used for early stopping during training. To ensure comparability, the energy price data were normalized between 0 and 1, using the maximum price of USD 190.81/MWh as the reference for normalization. Once the model has been trained, it can be applied as long as real-time energy price data are available. In cases where the dataset contains missing values, these can be handled using linear interpolation or more advanced methods such as autoencoders to reconstruct the missing data [32]. The simulation was

conducted using a 100 MWh battery, with the SoC at time slot $t = 0$ initialized to 0.5, representing 50% of the battery's capacity. To prevent battery degradation, the SoC_{min} and SoC_{max} values were set to 0.1 and 0.9, respectively. These settings allowed us to evaluate the performance of the reinforcement learning methods under realistic operational constraints while balancing long-term battery health and energy arbitrage profits. The constant battery cell parameters were referenced from [5]. Following the existing ESS energy arbitrage research, cumulative revenue was selected as the primary evaluation metric to effectively assess the performance gap between discrete and continuous RL approaches [4,5].

Table 1 summarizes the hyperparameters used in this experiment. Given the large number of hyperparameters that need to be predetermined for reinforcement learning models, it is nearly impossible to optimize all of them. Instead, we made small adjustments based on hyperparameters used in established benchmarks. The parameters for the LSTM network (number of layers and nodes) were based on those used in [4,5], the discrete reinforcement learning parameters were derived from [4], and the continuous RL parameters referenced the configurations used in [5]. When selecting the hyperparameters, we considered not only the performance of the models but also the learning speed to ensure efficient training. All networks were trained using the Adam optimizer [33], which is well-suited for handling the complexity of deep reinforcement learning. The entire framework was implemented in PyTorch 2.5 [34], utilizing Google Colab's GPU resources.

Table 1. Hyperparameters.

Hyperparameters	Value
The number of hidden layers in LSTM	2
The size of hidden neurons in LSTM	16
Learning rate	0.001
Discount factor (γ)	0.99
Minibatch size (discrete RL)	32
Size of experience replay buffer D (discrete RL)	50,000
Exploration rate (discrete RL)	0.1
The number of timesteps in episode (continuous RL)	128
Exploration standard deviation (continuous RL)	0.1

4.2. Results

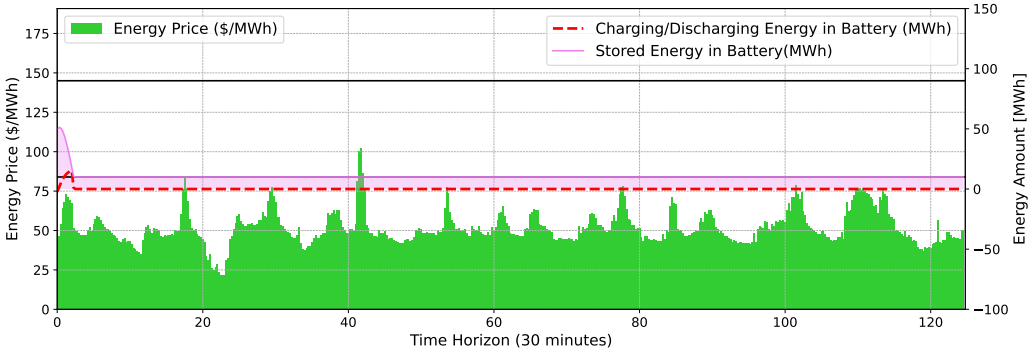
Table 2 compares the average profit per 30 min interval across the four models, as the dataset was sampled at 30 min intervals. This allowed for a consistent comparison of the results. The findings show that selecting an appropriate value for the Lagrangian multiplier ($\lambda = 0.1$) in continuous RL maximizes profits, as opposed to using values that are either too small ($\lambda = 0$) or too large ($\lambda = 1$). However, regardless of the λ value chosen, discrete RL outperformed continuous RL in terms of profitability. Although it might be expected that continuous RL would perform better, given that the charge and discharge amounts are naturally continuous variables, the experimental results show otherwise. Specifically, discrete RL demonstrated a 42% improvement over the best-performing continuous RL model, delivering significantly superior results. This surprising outcome suggests the need for a deeper analysis of the control strategies employed by the models. Understanding what led to these differences requires further investigation into how the models make decisions under various real-time energy price scenarios and how these decisions impact the SoC over time.

Table 2. Experiment results.

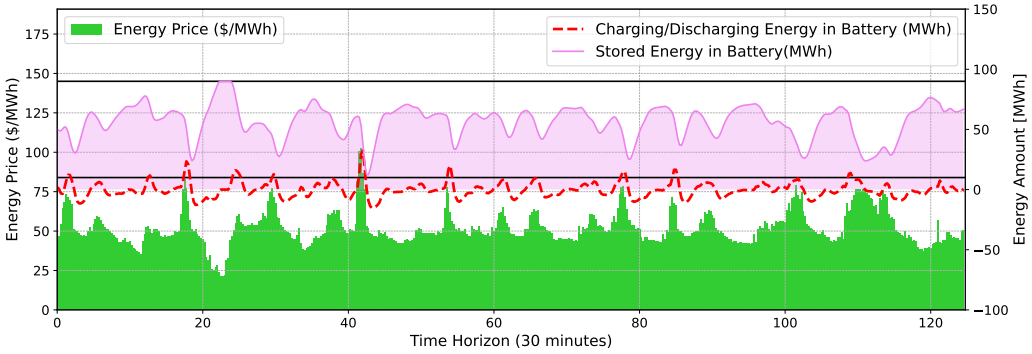
	30 min Averaged Profit (USD)
Continuous RL (λ = 0)	5.369
Continuous RL (λ = 1)	14.569
Continuous RL (λ = 0.1)	22.823
Discrete RL	32.392

Figure 5 illustrates the charge/discharge decisions and the corresponding changes in SoC based on real-time energy prices. The charging (−) and discharging (+) actions are represented by the red curve, the SoCs are depicted with the violet curve and filling, and the electricity prices are shown with the green bars. The black line marks the boundaries for the minimum and maximum stored energy, set at 0.1 and 0.9, respectively. The minimum and maximum stored energy limits influence the availability of the ESS, where a wider range can increase arbitrage profits, though these limits should be set carefully to minimize battery degradation. In the case of continuous RL with $\lambda = 0$, the model quickly discharges all the initially stored energy and takes no further actions, effectively missing out on future arbitrage opportunities. In contrast, the other models demonstrate more appropriate actions, charging when prices are low and discharging when prices are high. For continuous RL with $\lambda = 1$, the agent's actions are overly conservative, avoiding states where the SoC approaches its minimum or maximum limits. This results in suboptimal utilization of the energy storage system (ESS), as the model fails to engage in aggressive energy arbitrage. Setting λ to 0.1 was found to provide an optimal balance between energy arbitrage opportunities and SoC boundary constraints. This value allows the model to occasionally exceed SoC limits slightly, maximizing profit while effectively managing SoC boundaries through action clipping. The discrete RL model, however, stands out by primarily maintaining states of either fully charged or fully discharged SoC. This is because discrete RL restricts the agent's actions to either reaching the SoC limits or remaining idle. Paradoxically, this limitation appears to lead to more efficient utilization of the ESS for energy arbitrage. The optimal $\lambda = 0.1$ in continuous RL still struggles with fully leveraging the ESS due to the constraint imposed by the penalty mechanism, while discrete RL, unencumbered by such constraints, maximizes its use of the storage capacity. This further explains why discrete RL achieved superior performance in terms of profit compared with continuous RL across different configurations.

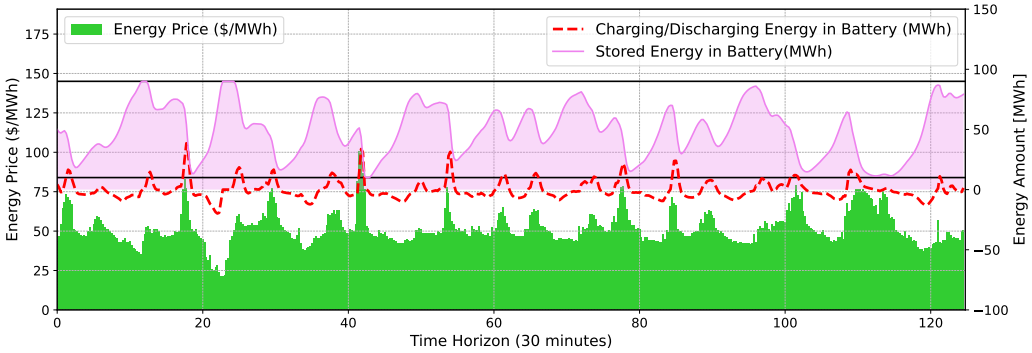
The cumulative profits of all methods over the entire test set are shown in Figure 6. In the case of continuous RL with $\lambda = 0$, the model initially achieves high profits by discharging and selling the all stored energy, but it fails to recharge and take further buying actions. As a result, although the initial profit is relatively high, the model is unable to generate additional revenue through energy arbitrage over time. In contrast, the remaining three models continue to generate profits steadily throughout the test period. When $\lambda = 0.1$, the model effectively balances immediate profit opportunities with SoC constraints, unlike $\lambda = 0$, which overly emphasizes short-term gains, or $\lambda = 1$, which limits arbitrage potential by overly restricting charge and discharge actions. Once again, discrete RL stands out with significantly higher profit generation, and the gap between discrete RL and the other models widens as time progresses. This suggests that discrete RL is consistently more effective at exploiting energy arbitrage opportunities. Figure 7 presents the cumulative distribution function of the 30 min revenue for all models. While the continuous RL methods, due to their finer control capabilities, were able to generate smaller incremental profits, the largest revenue gains were achieved by the discrete RL model. This again demonstrates the superior utilization of the ESS by discrete RL, as it capitalizes on larger arbitrage opportunities. The results suggest that, while continuous RL may offer more granular control over charge and discharge actions, discrete RL is better suited for maximizing profits through more aggressive and efficient ESS management.



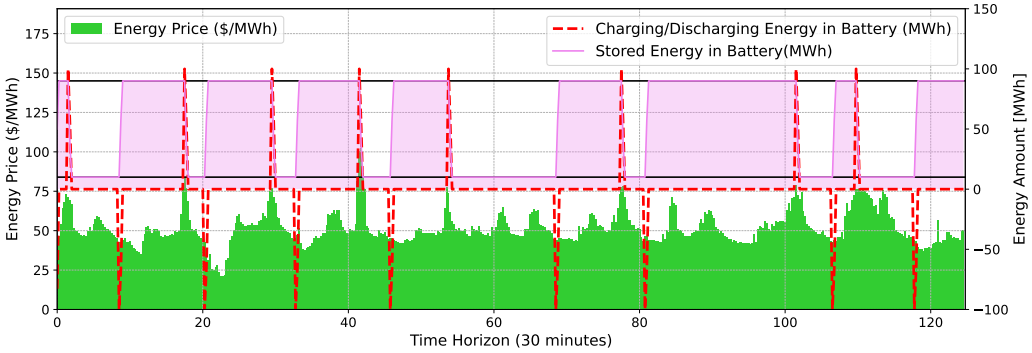
(**a**) Continuous RL ($\lambda = 0$).



(**b**) Continuous RL ($\lambda = 1$).



(**c**) Continuous RL ($\lambda = 0.1$).



(**d**) Discrete RL.

Figure 5. The charging/discharging results for four cases (green bar represents electricity prices; the red curve with the right axis represents the charging(−)/discharging(+) actions; the violet curve and filling represent the SoC; and the black line represents the minimum/maximum stored energy).

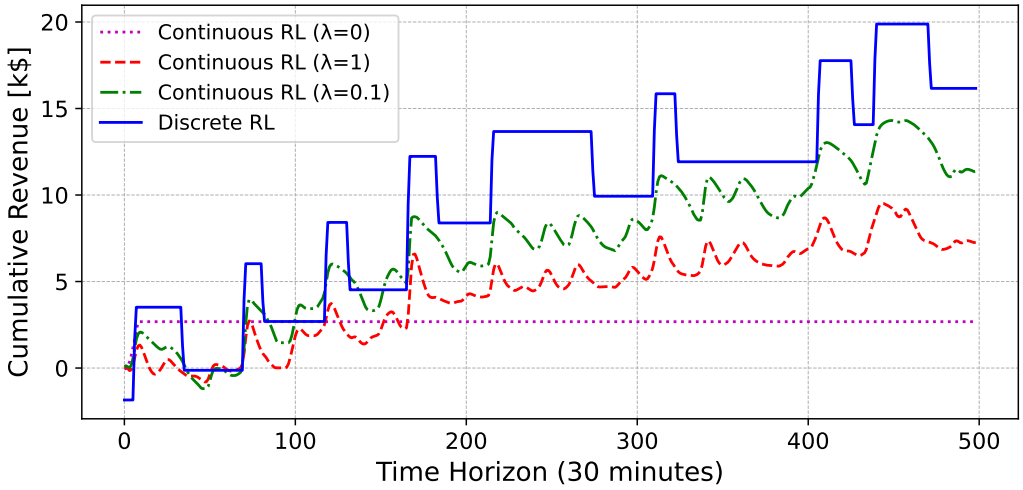


Figure 6. Comparison results of cumulative profits.

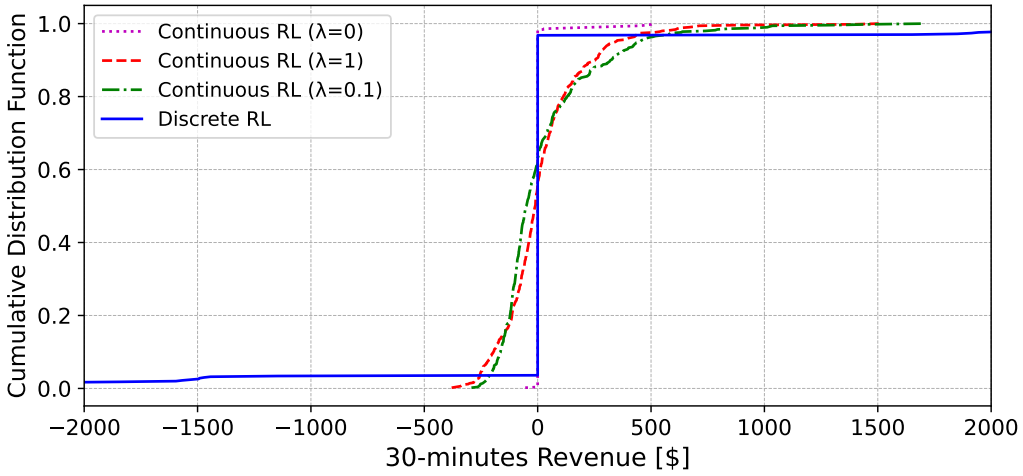


Figure 7. CDF of the 30 min profits.

While reaching fully charged or discharged states may not always be ideal, our results suggest that this approach is generally beneficial. Reflecting these benefits, other studies using discrete RL have also included fully charged or discharged states within the action space [4,11–13]. However, concerns may arise about its tendency to perform full charge and discharge cycles, potentially overusing the ESS and accelerating its degradation. Since frequent charge/discharge cycles shorten the ESS lifespan, it is essential to examine the cumulative charge and discharge volumes over the entire dataset. Figure 8 presents these findings. Contrary to expectations, continuous RL with $\lambda = 0.1$ had the highest cumulative ESS usage, followed by discrete RL, with continuous RL using $\lambda = 1$ closely behind. When $\lambda = 1$, the charge/discharge volume was only 18% lower than when $\lambda = 0.1$, yet the profit was reduced by 36%, indicating that the policy with $\lambda = 1$ was overly conservative. The reason discrete RL did not have the highest usage is that it often opted for idle actions instead of smaller charge or discharge operations. In real-world energy arbitrage applications, it would be necessary to impose constraints on ESS charge/discharge volumes to avoid excessive wear. As shown in Figure 8, the lifespan impact of discrete RL is comparable to that of continuous RL, indicating that the performance advantage of discrete RL does not come at the cost of increased ESS degradation.

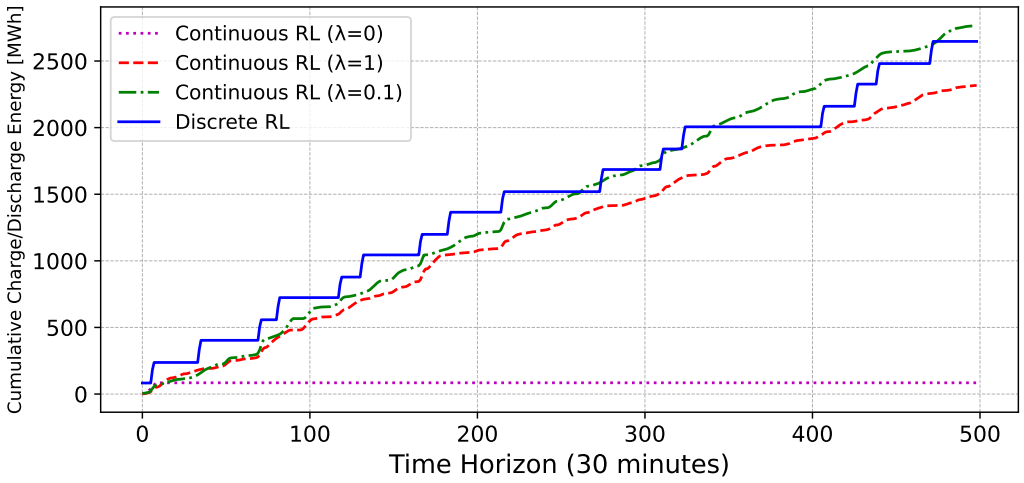


Figure 8. Comparison results of cumulative charge/discharge energy.

5. Conclusions

In conclusion, this study has demonstrated that discrete RL outperforms continuous RL in energy arbitrage with ESS, offering clearer insights into why discrete RL is commonly preferred in recent research. Continuous RL, while theoretically offering finer control over charging and discharging actions, faces significant challenges due to the need for action clipping at the boundaries of the SoC. This clipping often results in actions becoming stuck at the SoC limits, impairing the learning process and leading to suboptimal performance. Even with the implementation of constrained RL techniques, which are designed to manage such boundaries, the resulting policies tend to be overly conservative, limiting the full utilization of the ESS. On the other hand, discrete RL, despite its coarser control of actions, avoids these pitfalls by naturally guiding the ESS towards fully charged or fully discharged states. This leads to more effective utilization of the storage system and better overall performance in energy arbitrage tasks. Our simulations confirmed that discrete RL consistently achieves higher profits while maintaining manageable levels of ESS usage, highlighting its practicality and effectiveness.

With current techniques, discrete RL has proven to be simpler and more effective than continuous RL for the energy arbitrage problem. However, this does not imply that discrete RL should remain the default choice for energy arbitrage in the future. Rather than simply substituting discrete RL with continuous RL, future research should focus on developing methods that address the limitations of continuous RL. The goal should be to create RL techniques that retain the strengths of discrete RL while incorporating the fine-grained control capabilities of continuous RL. In future studies, the development of such hybrid approaches could lead to even greater performance improvements, combining the best of both methods to enhance the efficiency and effectiveness of energy arbitrage. Additionally, we expect that this approach could be scaled to more complex systems, such as those with multiple ESS units or hybrid configurations incorporating hydrogen or thermal storage, making this a promising area for future research. Furthermore, we suggest exploring the adaptability of RL to more complex and unpredictable systems, such as energy price spikes or rapid changes in renewable output, as a promising direction for future research. With well-designed neural network architectures, such as transformers, RL has the potential to effectively manage even more unpredictable and complex scenarios.

Author Contributions: Conceptualization, J.J., T.-Y.K. and W.-K.P.; methodology, J.J.; software, J.J.; validation, J.J., T.-Y.K. and W.-K.P.; writing—original draft preparation, J.J.; writing—review and editing, J.J.; visualization, J.J.; supervision, W.-K.P.; project administration, T.-Y.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) and the Ministry of Trade, industry & Energy (MOTIE) of Korea (No. 2021202090028C).

Data Availability Statement: The data presented in this study are openly available [31].

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ESS	Energy storage system
SoC	State of charge
RL	Reinforcement learning
DQN	Deep Q-network
A2C	Advantage actor–critic
PPO	Proximal policy optimization
TinyML	Tiny machine learning

References

1. Vejdan, S.; Grijalva, S. The value of real-time energy arbitrage with energy storage systems. In Proceedings of the 2018 IEEE Power & Energy Society General Meeting (PESGM), Portland, OR, USA, 5–10 August 2018; IEEE: Piscataway Township, NJ, USA, 2018; pp. 1–5.
2. Mikkelson, D.; Frick, K. Analysis of controls for integrated energy storage system in energy arbitrage configuration with concrete thermal energy storage. *Appl. Energy* **2022**, *313*, 118800. [CrossRef]
3. Khakimov, R.; Moskvin, A.; Zhdaneev, O. Hydrogen as a key technology for long-term & seasonal energy storage applications. *Int. J. Hydrog. Energy* **2024**, *68*, 374–381.
4. Cao, J.; Harrold, D.; Fan, Z.; Morstyn, T.; Healey, D.; Li, K. Deep Reinforcement Learning-Based Energy Storage Arbitrage With Accurate Lithium-Ion Battery Degradation Model. *IEEE Trans. Smart Grid* **2020**, *11*, 4513–4521. [CrossRef]
5. Jeong, J.; Kim, S.W.; Kim, H. Deep reinforcement learning based real-time renewable energy bidding with battery control. *IEEE Trans. Energy Mark. Policy Regul.* **2023**, *1*, 85–96. [CrossRef]
6. Chakraborty, T.; Watson, D.; Rodgers, M. Automatic Generation Control Using an Energy Storage System in a Wind Park. *IEEE Trans. Power Syst.* **2018**, *33*, 198–205. [CrossRef]
7. Hashmi, M.U.; Mukhopadhyay, A.; Bušić, A.; Elias, J.; Kiedanski, D. Optimal storage arbitrage under net metering using linear programming. In Proceedings of the 2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), Beijing, China, 21–24 October 2019; IEEE: Piscataway Township, NJ, USA, 2019; pp. 1–7.
8. Sutton, R.S.; Barto, A.G. *Introduction to Reinforcement Learning*; MIT Press: Cambridge, UK, 1998; Volume 135.
9. Jeong, J.; Ku, T.Y.; Park, W.K. Time-Varying Constraint-Aware Reinforcement Learning for Energy Storage Control. *arXiv* **2024**, arXiv:2405.10536.
10. Miao, Y.; Chen, T.; Bu, S.; Liang, H.; Han, Z. Co-optimizing battery storage for energy arbitrage and frequency regulation in real-time markets using deep reinforcement learning. *Energies* **2021**, *14*, 8365. [CrossRef]
11. Madahi, S.S.K.; Claessens, B.; Develder, C. Distributional Reinforcement Learning-based Energy Arbitrage Strategies in Imbalance Settlement Mechanism. *arXiv* **2023**, arXiv:2401.00015.
12. Karimi madahi, S.s.; Gokhale, G.; Verwee, M.S.; Claessens, B.; Develder, C. Control Policy Correction Framework for Reinforcement Learning-based Energy Arbitrage Strategies. In Proceedings of the 15th ACM International Conference on Future and Sustainable Energy Systems, Singapore, 4–7 June 2024; pp. 123–133.
13. Harrold, D.J.; Cao, J.; Fan, Z. Data-driven battery operation for energy arbitrage using rainbow deep reinforcement learning. *Energy* **2022**, *238*, 121958. [CrossRef]
14. Achiam, J.; Held, D.; Tamar, A.; Abbeel, P. Constrained policy optimization. In Proceedings of the International Conference on Machine Learning. PMLR, Sydney, NSW, Australia, 6–11 August 2017; pp. 22–31.
15. Liang, Q.; Que, F.; Modiano, E. Accelerated primal-dual policy optimization for safe reinforcement learning. *arXiv* **2018**, arXiv:1802.06480.
16. Lee, S.; Choi, D.H. Federated reinforcement learning for energy management of multiple smart homes with distributed energy resources. *IEEE Trans. Ind. Inform.* **2020**, *18*, 488–497. [CrossRef]
17. da Silva André, J.; Stai, E.; Stanojev, O.; Hug, G. Battery control with lookahead constraints in distribution grids using reinforcement learning. *Electr. Power Syst. Res.* **2022**, *211*, 108551. [CrossRef]
18. Park, S.; Pozzi, A.; Whitmeyer, M.; Perez, H.; Kandel, A.; Kim, G.; Choi, Y.; Joe, W.T.; Raimondo, D.M.; Moura, S. A deep reinforcement learning framework for fast charging of Li-ion batteries. *IEEE Trans. Transp. Electrif.* **2022**, *8*, 2770–2784. [CrossRef]
19. Hesse, H.C.; Kumtepeli, V.; Schimpe, M.; Reniers, J.; Howey, D.A.; Tripathi, A.; Wang, Y.; Jossen, A. Ageing and efficiency aware battery dispatch for arbitrage markets using mixed integer linear programming. *Energies* **2019**, *12*, 999. [CrossRef]
20. Cheng, B.; Powell, W.B. Co-optimizing battery storage for the frequency regulation and energy arbitrage using multi-scale dynamic programming. *IEEE Trans. Smart Grid* **2016**, *9*, 1997–2005. [CrossRef]

21. Zheng, N.; Jaworski, J.; Xu, B. Arbitraging variable efficiency energy storage using analytical stochastic dynamic programming. *IEEE Trans. Power Syst.* **2022**, *37*, 4785–4795. [CrossRef]
22. Seyde, T.; Gilitschenski, I.; Schwarting, W.; Stellato, B.; Riedmiller, M.; Wulfmeier, M.; Rus, D. Is bang-bang control all you need? solving continuous control with bernoulli policies. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 27209–27221.
23. Abed, A.M.; Mouziraji, H.R.; Bakhshi, J.; Dulaimi, A.; Mohammed, H.I.; Ibrahem, R.K.; Ben Khedher, N.; Yaïci, W.; Mahdi, J.M. Numerical analysis of the energy-storage performance of a PCM-based triplex-tube containment system equipped with arc-shaped fins. *Front. Chem.* **2022**, *10*, 1057196. [CrossRef]
24. Chen, M.; Rincon-Mora, G. Accurate electrical battery model capable of predicting runtime and I-V performance. *IEEE Trans. Energy Convers.* **2006**, *21*, 504–511. [CrossRef]
25. Morstyn, T.; Hredzak, B.; Aguilera, R.P.; Agelidis, V.G. Model Predictive Control for Distributed Microgrid Battery Energy Storage Systems. *IEEE Trans. Control Syst. Technol.* **2018**, *26*, 1107–1114. [CrossRef]
26. Ma, Y.J.; Shen, A.; Bastani, O.; Dinesh, J. Conservative and adaptive penalty for model-based safe reinforcement learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 22 February–1 March 2022; Volume 36, pp. 5404–5412.
27. Huang, S.; Kanervisto, A.; Raffin, A.; Wang, W.; Ontañón, S.; Dossa, R.F.J. A2C is a special case of PPO. *arXiv* **2022**, arXiv:2205.09123.
28. De La Fuente, N.; Guerra, D.A.V. A Comparative Study of Deep Reinforcement Learning Models: DQN vs PPO vs A2C. *arXiv* **2024**, arXiv:2407.14151.
29. Troudi, F.; Jouini, H.; Mami, A.; Ben Khedher, N.; Aich, W.; Boudjemline, A.; Boujelbene, M. Comparative assessment between five control techniques to optimize the maximum power point tracking procedure for PV systems. *Mathematics* **2022**, *10*, 1080. [CrossRef]
30. Kuppusamy, P.; Kapadia, D.; Manvitha, E.G.; Dhahbi, S.; Iwendi, C.; Khan, M.I.; Mohanty, S.N.; Khedher, N.B. EL-RFHC: Optimized ensemble learners using RFHC for intrusion attacks classification. *Ain Shams Eng. J.* **2024**, *15*, 102807. [CrossRef]
31. The Changing Price of Wholesale UK Electricity over More Than a Decade. 2017. Available online: https://www.ice.org.uk/knowledge-and-resources/briefing-sheet/the-changing-price-of-wholesale-uk-electricity (accessed on 25 October 2024).
32. Jeong, J.; Ku, T.Y.; Park, W.K. Denoising Masked Autoencoder-Based Missing Imputation within Constrained Environments for Electric Load Data. *Energies* **2023**, *16*, 7933. [CrossRef]
33. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
34. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 8026–8037.