Linear Reinforcement Learning for Energy Storage Systems Optimal Dispatch

Shuyi Gao[1], Shengren Hou[1], Edgar Mauricio Salazar Duque[2], Peter Palensky[1], Pedro P. Vergara[1]
[1]Intelligent Electrical Power Grids, Delft University of Technology, Delft, The Netherlands
[2]Energy Systems Systems Group, Eindhoven University of Technology, Eindhoven, The Netherlands
s.gao@tudelft.nl, h.hou-1@tudelft.nl, e.m.salazar.duque@tue.nl, p.palensky@tudelft.nl, p.p.vergarabarrios@tudelft.nl

***Abstract*—Reinforcement Learning (RL) has emerged as a promising solution for defining the optimal dispatch of Energy Storage Systems (ESS) in distributed energy systems. However, a notable gap exists in the literature: a lack of comprehensive and fair comparisons between different RL algorithms, particularly between linear and nonlinear approaches. This study critically evaluates the trade-offs between computational efficiency and operational accuracy among various Linear RL (LRL) strategies and compares them against the nonlinear Deep-Q-Network (DQN) algorithm. Through a comprehensive analysis, this study benchmarks the model-based Mixed-Integer Linear Programming (MILP) results to assess and compare these algorithms' convergence, training efficiency, and optimization accuracy. Results indicate that while LRL approaches the operational cost accuracy of DQN, it faces significant trade-offs in computational efficiency and struggles with generalization across larger and varied datasets. The results illuminate critical areas for further development in LRL methodologies, particularly in enhancing their adaptability and generalization capabilities.**

***Index Terms*—Reinforcement Learning, Linear Representation, Neural Network, Energy Storage System, Optimal Dispatch**

NOTATION

Sets	
$\mathcal{B}$	set of ESS
$\mathcal{V}$	set of PVs
$\mathcal{L}$	set of load demands
$\mathcal{T}$	set of time steps of MDP
$\mathcal{S}$	set of states of MDP
$\mathcal{A}$	set of actions of MDP
$\mathcal{P}$	set of transition probabilities of MDP
$\mathcal{R}$	set of rewards of RL
Indexes	
i	the i th battery of ESS, $i \in \mathcal{B}$
j	the j th PV generator, $j \in \mathcal{V}$
k	the k th load demand unit, $k \in \mathcal{L}$
t	time step $t \in \mathcal{T}$
Parameters	
$\overline{P_i^B}$, $\underline{P_i^B}$	the i th battery's maximum/minimum charging/discharging power
$\overline{E_i^B}$, $\underline{E_i^B}$	the i th battery's maximum/minimum SOC
η_i	the i th battery's energy efficiency
μ_i	the i th battery's discretization parameter
Δt	the discretization time step
σ_t	the price of electricity at time step t
φ_0, φ_1	the control factor of reward function
α	the learning rate of RL algorithms
γ	the discount factor of MDP
λ	the decay rate of eligibility traces
τ	the update rate of DQN's target network
Variables	
$P_{i,t}^B$	the i th battery's active power at time step t
$SOC_{i,t}^B$	the i th battery's SOC at time step t
$P_{j,t}^V$	the j th PV's active power at time step t
$P_{k,t}^L$	the k th load demand unit's active power at time step t

This work is supported by the Chinese Scholarship Council.

I. INTRODUCTION

The increasing penetration of Renewable Energy Sources (RES), such as Photovoltaics (PV) and Wind Turbines (WT), introduces significant complexity and volatility into the distribution networks due to their intermittent nature. This variability poses critical challenges for maintaining stable and economical operations, particularly in balancing supply and demand effectively [1]. Energy Storage Systems (ESS) play a pivotal role in addressing these challenges. With their capability to store excess energy during periods of high renewable output or low electricity prices and release it during peak demand or low generation periods, ESSs enhance energy supply's consistency, reliability, and economic efficiency [2].

Various model-based and model-free optimization algorithms have been explored to solve the ESS optimal dispatch problem. Model-based methods, including stochastic and robust optimization, have shown success in addressing the optimal dispatch of ESSs [3–6]. However, these methods often rely on accurate, complete knowledge of the operational environment [7]. Furthermore, model-based approaches tend to be computationally intensive, particularly when dealing with a large number of scenarios, which may lead to conservative and time-consuming optimization results [8].

In contrast, optimal energy dispatch problems can effectively be transformed into a Markov Decision Process (MDP) that can be solved by using model-free Reinforcement Learning (RL) [9]. RL algorithms leverage historical data to iteratively improve control decisions through repeated interactions with the environment without the need for prior system models [8]. The growing availability of data from smart meters and sensors further enhances the viability of data-driven RL approaches[10]. Traditional RL methods, such

as Q-Learning, can find optimal solutions with discrete state and action pairs by updating an optimal Q-table. To manage larger or continuous spaces, the action-value functions are typically represented using parameterized forms such as linear functions, i.e., Polynomial (Poly), Fourier series (Four), Radial Base Function (RBF), Tile Coding (TC), or nonlinear neural networks (NN) [11]. In this concept, RL algorithms using linear feature functions to represent a state can be categorized as Linear RL (LRL) algorithms, while those employing NN as Non-Linear RL (NonLRL) algorithms.

Based on Q-Learning with NN, namely Deep Q-Learning (DQN), [12] proposed a NoisyNet-DoubleDQN to learn the optimal strategy for energy storage participating in the energy arbitrage, it was validated against the MILP model using actual electricity prices, showing its effectiveness. [13] compared four advanced NonLRL for managing generators and energy storage in energy system scheduling. The results highlight that by leveraging NN, RL agents can deliver high-quality, real-time solutions, even in unseen scenarios, exhibiting NN's strong generalization and the ability to handle complex, high-dimensional nonlinear relationships. However, these models do not inherently guarantee convergence to optimal solutions, meaning achieving the optimal solution is not ensured. LRL algorithms are mathematically proven to be convergence-guaranteed; in [14] and [15], LRL with TC and RBF are implemented for battery strategy management and load tap setting to assure an economical and reliable energy system operation, showing that they are capable of approaching the optimal solutions provided by MILP model. Evidence from sequential tasks in environments such as OpenAI Gym and Mujoco has demonstrated that LRL can achieve comparably high or even superior performance, challenging the traditional reliance on NNs for complex decision-making tasks [16, 17]. Despite the capabilities of RL in different applications, there remains a gap in the literature regarding a fair comparison of LRL and NonLRL algorithms in solving the optimal energy dispatch of ESSs.

Our study addresses the existing gaps in the comparison of LRL and NonLRL for optimal energy dispatch in ESS. By employing MILP results as a benchmark, we aim to rigorously assess and compare the performance of LRL algorithms, specifically LRL-Poly, LRL-Four, LRL-RBF, and LRL-TC with the nonlinear DQN. Our evaluation focuses on three key aspects: convergence rates, training efficiency, and optimization accuracy. Through detailed sensitivity analysis, we explored the impact of tiles and tilings configurations on LRL performance. Additionally, using various datasets simulating different operation scenarios, we tested the adaptability of both LRL and DQN to dynamic changes in demand, supply, and pricing, gaining insights into their generalizability.

II. Optimal Dispatch of ESSs as an MDP Problem

A distribution system composed of ESSs, PVs, load demand units, and an external power grid connection. The objective of the optimal ESSs dispatch problem is to minimize the total operational costs of the systems throughout the entire day by strategically defining the charging and discharging actions of ESS at each time step within the day.

Given this nature, this problem is classified as a sequential decision-making challenge, primarily due to the time-dependent behavior of the ESS's state-of-charge (SOC). The SOC is directly changed by the charging/discharging operations defined in the preceding interval. Furthermore, the decision is made based on time-varying uncertainties, such as fluctuations in RES generation, variations in load demand, and dynamic electricity prices. The mathematical formulation of the optimal ESS dispatch problem is presented below:

$$\min_{\sigma_t, P^B_{i,t}, P^V_{j,t}, P^L_{k,t}} \sum_{t\in\mathcal{T}} \sigma_t \Big(\sum_{k\in\mathcal{L}} P^L_{k,t} - \sum_{i\in\mathcal{B}} P^B_{i,t} - \sum_{j\in\mathcal{V}} P^V_{j,t}\Big)\Delta t \quad (1)$$

$$P^N_t = \sum_{k\in\mathcal{L}} P^L_{k,t} - \sum_{i\in\mathcal{B}} P^B_{i,t} - \sum_{j\in\mathcal{V}} P^V_{j,t}, \forall t \in \mathcal{T} \quad (2)$$

$$\underline{P^G_t} \le P^N_t \le \overline{P^G_t}, \forall t \in \mathcal{T} \quad (3)$$

$$\underline{P^B_i} \le P^B_{i,t} \le \overline{P^B_i}, \forall i \in \mathcal{B}, \forall t \in \mathcal{T} \quad (4)$$

$$\underline{E^B_i} \le SOC^B_{i,t} \le \overline{E^B_i}, \forall i \in \mathcal{B}, \forall t \in \mathcal{T} \quad (5)$$

$$SOC^B_{i,t} = SOC^B_{i,t-1} + \eta^B_i P^B_{i,t}, \forall i \in \mathcal{B}, \forall t \in \mathcal{T} \quad (6)$$

In this formulation, the objective function (1) seeks to minimize the total operational costs by defining the charge/discharge power $P^B_{i,t}$ of the ESS. P^N_t is the net load that is subject to (2) and is constrained by the limitations as in (3), which describes the power import/export limits within the main grid. Notably, when $P^N_t > 0$, power is fed into the main grid at the sell price $\sigma_t = \sigma_{s,t}$, while when $P^N_t < 0$, power is imported from the grid at a buy price $\sigma_t = \sigma_{b,t}$. $\overline{P^G_t}$ and $\underline{P^G_t}$ are the maximum import/export capacity between the system and the main grid. (4) and (5) defines ESS charge/discharge power and the SOC, respectively, where $P^B_{i,t} > 0$ means discharge, and $P^B_{i,t} < 0$ means charge. (6) models the evolution of the ESS's SOC after defining the charge/discharge actions.

The above-present ESS optimal energy dispatch problem can be formulated as an MDP described by a 5-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is the state space and $\mathcal{A}$ is the action space. $\mathcal{P}$ is the set of transition probabilities of the MDP where $\mathcal{P} : s, a \in \mathcal{S} \times \mathcal{A} \to p \doteq p(s^{'}|s, a)$ is the probability of state s taking action a transition to state $s^{'}$. $\mathcal{R}$ is the set of rewards where $\mathcal{R} : s, a, s^{'} \in \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to r \doteq \mathcal{R}(s, a, s^{'}) \in \mathbb{R}$ evaluating the effectiveness of action a at state s. $\gamma \in [0, 1]$ is the discount factor that weights rewards in the long term.

$$s_t = \{t, \sigma_t, P^N_t, SOC^B_{1,t}, SOC^B_{2,t}, \cdots, SOC^B_{i,t}\}, s_t \in \mathcal{S} \quad (7)$$

$$a_t = \{P^B_{1,t}, P^B_{2,t}, \cdots, P^B_{i,t}\}, a_t \in \mathcal{A} \quad (8)$$

The state at each time step s_t represents a comprehensive encapsulation of the current operational dynamics within the distributed energy system, which is defined by 7. The action space in (8) can be discretized by an increment of charging/discharging power i.e., each $P^B_{i,t} \in \{\underline{P^B_i}, \cdots, -2\Delta P^B_i, -\Delta P^B_i, 0, \Delta P^B_i, 2\Delta P^B_i, \cdots, \overline{P^B_i}\}$,

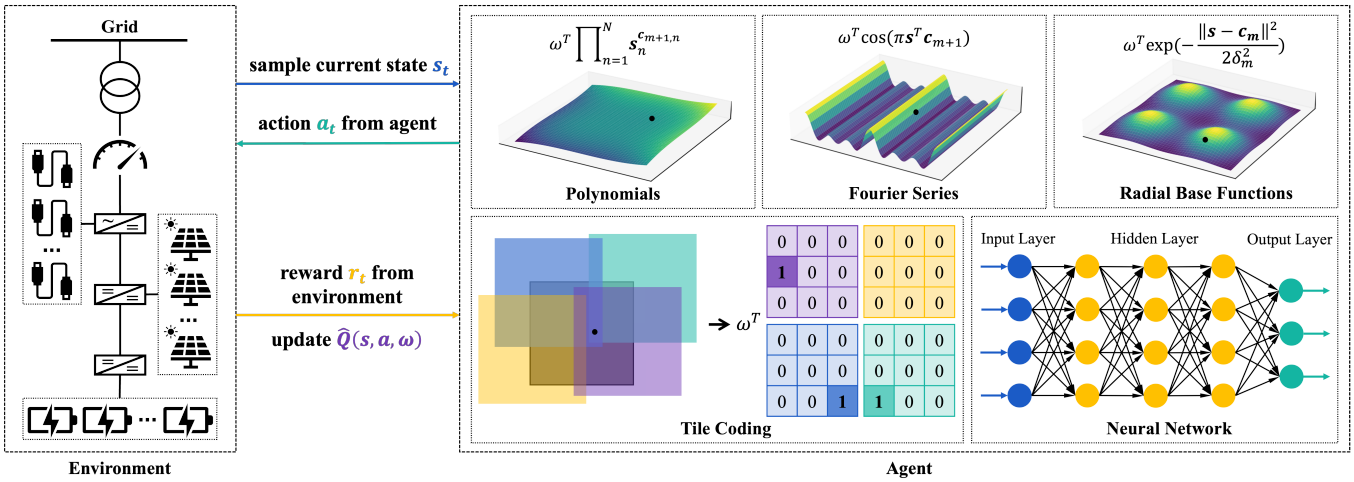

Fig. 1. Framework of training linear and nonlinear RL algorithms in solving the ESS optimal energy dispatch problem.

where $\Delta P_i^B = \frac{1}{\mu_i - 1}(\overline{P_i^B} - \underline{P_i^B}), \mu_i \in \mathcal{N}_+$. It is convenient to have a symmetric range of power levels, positive and negative values available to charge/discharge, and one option to set the battery on idle mode [14].

The reward is designed to guide RL algorithms learning an optimal policy, e.g., give a low value when agents define uneconomic and unsatisfactory decisions as ΔP_t in (9) exists, where $\hat{P_t^N}$ represents the actual power imported from the main grid when it is positive that objects to (10), verse visa. Thus, in the formulated MDP, the reward function in (11) contains two components that stimulate the RL agent to define actions that can synchronously consider the operational cost and power unbalance constraints, where φ_0 and φ_1 control the weight of these two goals.

$$\Delta P_t = |\sum_{k\in\mathcal{L}} P_{k,t}^L - \sum_{i\in\mathcal{B}} P_{i,t}^B + \sum_{j\in\mathcal{V}} P_{j,t}^V - \hat{P_t^N}|, \forall t \in \mathcal{T} \quad (9)$$

$$\hat{P_t^N} = \begin{cases} \overline{P_t^G}, & P_t^N > \overline{P_t^G} \\ P_t^N, & \underline{P_t^G} \le P_t^N \le \overline{P_t^G} \\ \underline{P_t^G}, & P_t^N < \underline{P_t^G} \end{cases} \quad (10)$$

$$r_t = \varphi_0 \left[-\sigma_t (\sum_{k\in\mathcal{L}} P_{k,t}^L - \sum_{i\in\mathcal{B}} P_{i,t}^B + \sum_{j\in\mathcal{V}} P_{j,t}^V) \right] + \varphi_1(\Delta P_t) \quad (11)$$

Linear and nonlinear approximation techniques are leveraged to overcome the limitations of tabular Q-representation within the continuous state space. Linear approaches, Poly, Four, RBF, and TC approximate the state space by computing the sum of weighted features. Poly and Four representations are generated by constructing m-th order cross-terms for each dimension n, facilitating the capture of interactions among various state variables. RBF features employ m Gaussian centers per dimension. Their effectiveness is contingent on the relative distance between a state s and a center c_m, adjusted according to the width δ_m, thereby localizing features around these centers. TC, alternatively, segments the state space into a grid of tiles across multiple overlapping tilings. Each tile acts as a binary feature that is activated (assigned a value of 1) when a state falls within its bounds, while all other tiles are deactivated (set to 0). Nonlinear approaches utilize an NN to represent the Q table, directly building the mapping between state-action pairs and a Q-value.

The training of these RL algorithms revolves around learning an optimal Q-function $Q^*(s,a)$ that satisfies the Bellman optimality equation, subsequently deriving the optimal policy implicitly. Both linear and nonlinear algorithms update the parameterized Q-function $\hat{Q}(s,a,\omega)$ towards the target Q-value by minimizing the estimation error. This process involves adjusting parameters to align closely with the Temporal-Difference (TD) target, substituting for the unknown real value of $Q(s,a)$. The Q-function updating uses gradient descent by sampling from the stationary distributions $p_\pi(s_d)$ and the replay buffer $\mathcal{D} = \{(s,a,s',r) : s,s' \in \mathcal{S}, a \in \mathcal{A}\}$. Following policy π, each state s_d has a certain probability $p_\pi(s_d)$ of being visited while the replay buffer stores and replays the trajectories collected from the interactions with the environment. The update function can be described as follows:

$$\begin{aligned} \omega_{t+1} &= \omega_t + \alpha(Q(s,a) - \hat{Q}(s,a,\omega_t))\nabla\hat{Q}(s,a,\omega_t) \\ &= \omega_t + \alpha(r_t + \gamma\hat{Q}(s',a',\omega_t)) - \hat{Q}(s,a,\omega_t))\nabla\hat{Q}(s,a,\omega_t) \end{aligned} \quad (12)$$

III. Case Study

In this paper, the time interval between charge/discharge decisions is set to an hour, and for each episode, $\mathcal{T} = 24$. The selling price $\sigma_{s,t} = 0.5\sigma_{b,t}$ is set, where $\sigma_{b,t}$ is the price of import electricity from the main grid at time t. The exchanging ability of the grid is set as $400kW$. For ESS parameters, the capacity of the ESS is set as $200kWh$, the maximum and minimum charging/discharging power is set as $[-80, 80]kW$, the SOC is limited within $[0.2, 0.8]$, and the energy efficiency is set as 1. The action space is discretized as $\mu = 9$, and the initial SOC of each episode during training is set as 0.2. The parameters φ_0 and φ_1 are set as 0.5 and 50, respectively.

Leveraging the ϵ-greedy policy [11] during training so the exploration and exploitation can be well balanced, and $\epsilon = 1$ is initialized. The decay rate of ϵ is set at 0.9996 for RL algorithms, so each gets the same opportunity to explore potential high-reward actions at the beginning of training. 5 order polynomials and Fourier series, 6 RBF centers, and 4 tiles in each dimension are implemented in linear RL agents. The SARSA [14] with eligibility traces is implemented with the four linear feature representations, in which $\lambda = 0.9$ is set, while the DQN target network's update rate τ is set as 0.005. All algorithms are implemented in Python, and the MILP optimization problem is formulated and solved by Pyomo packages as benchmark results.

The performance of RL algorithms is evaluated using two key indicators: training duration and operational cost error. Operational cost error is determined by comparing results from DRL algorithms against the global optimal solution obtained from solving MILP. Training duration refers to the computational time for training RL agents until convergence (10000 episodes). Average results from five random seed simulations are used to eliminate the randomness. Implemented details of the environment and RL algorithms are open-sourced in [18].

A. *Performance of LRLs on 1-day Operation*

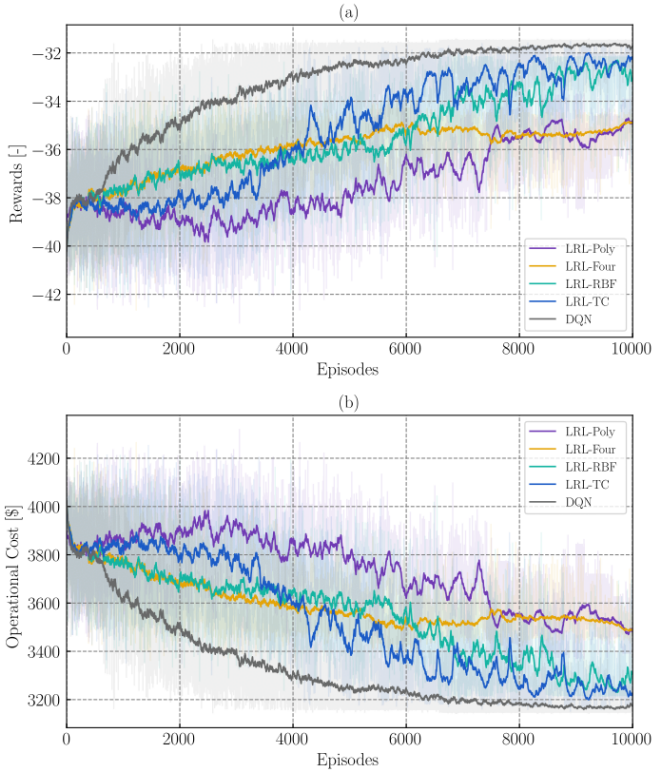

Fig. 2. Training results of RL agents on one-day operation during 10000 episodes, (a) rewards, (b) operation costs.

Fig. 2 shows the average reward and operational cost of all RL algorithms over 10000 episodes. During the first 500 episodes, RL agents generally selected actions yielding rewards between -40 and -38. Over successive episodes, DQN demonstrated a consistent increase, surpassing all the LRLs with an average reward close to -31.66; both LRL-TC and LRL-RBF exhibited rising rewards with terminal average values close to DQN's, ranging from -33.10 to -32.17, which implies their comparable performance to DQN. LRL-Poly exhibited modest gains after the early episodes and maintained steady thereafter. LRL-Four's rewards improved from around -38 to -36 by episode 5000 but failed to maintain this growing pattern and eventually converged at an average reward of around -35.42, a lower optimal value than other algorithms. Operational costs showed a near-symmetrical decreasing pattern compared to those rewards, reflecting the agents' successful adherence to power imbalance constraints.

Fig. 3 illustrates the operational day's dynamics, presenting PV generation, load demand, and electricity pricing alongside the evolving SOC of the ESS, as governed by various agents' learned charging/discharging strategies. Fig. 3 (b)-(f) presents

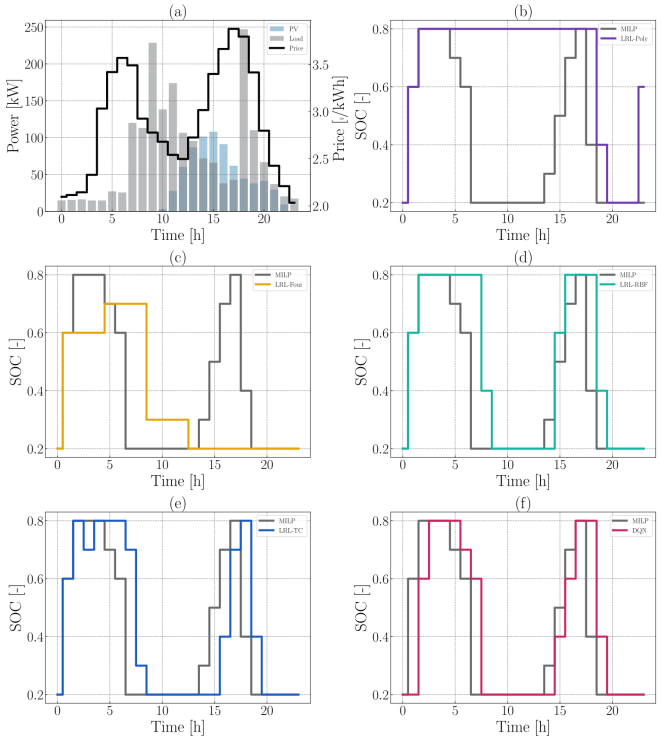

Fig. 3. Comparison of RL agents and MILP on one-day operation, (a) PV generation, load consumption, and electricity price, (b) - (f) comparison of ESS's SOC changing of the given charging and discharging operation strategy of LRL-Poly, LRL-Four, LRL-RBF, LRL-TC, DQN, and MILP.

the strategy derived from MILP. It involves initiating charging at midnight to exploit low prices and demand, reaching maximum SOC by 2:00, which it maintains until 4:00 to coincide with a price surge, and optimizing profit by discharging at peak rates. The SOC hits its minimum by 14:00, just as PV output exceeds load, preceding another price spike, making it economical to resume storing energy. At 18:00, in response to a load spike, the system discharges and then stabilizes at minimum SOC from 19:00 onwards. Among the agents,

TABLE I

MEAN AND 95% CONFIDENCE BOUNDS OF TRAINING DURATION [s], OPERATIONAL COST [$], AND ERROR [%] OF RL ALGORITHMS ON ONE-DAY OPERATION

Algorithms	**LRL-Poly**	**LRL-Four**	**LRL-RBF**	**LRL-TC**	**DQN**	**MILP**
Total duration [s]	3082.96 (3077.15, 3088.76)	4816.95 (4810.59, 4823.30)	2411.11 (2408.55, 2413.67)	**107.42** **(107,10, 107.74)**	618.62 (617.14, 620.10)	-
Operation costs [$]	3541.36 (3495.98, 3586.74)	3542.65 (3524.29, 3561.01)	3283.47 (3256.54, 3310.40)	**3234.26** **(3217.72, 3250.80)**	3166.84 (3159.39, 3174.30)	3142.40
Error [%]	12.70 (11.25, 14.14)	12.74 (12.15, 13.32)	4.49 (3.63, 5.35)	**2.92** **(2.40, 3.45)**	0.78 (0.54, 1.02)	-

the LRL-Poly recognized only the initial and final phases, resulting in a high operational cost of 3424.70. LRL-Four identified the first charging phase but missed subsequent discharging opportunities, costing 3479.35. In contrast, LRL-RBF, LRL-TC, and DQN closely followed MILP's strategy with minor timing deviations. DQN aligned most closely with MILP, achieving a cost of 3147.65. LRL-RBF, with slightly delayed actions, incurs 3215.11, while LRL-TC's delayed charging effectively uses PV output, costing 3176.16.

The disparities in RL agent performance are due to the nature of their function approximators and their interaction with the complex state-action space. DQN excels with its NN architecture, which adeptly handles nonlinear patterns, whereas LRL-TC and LRL-RBF effectively generalize strategies through their approximation functions. However, LRL-Poly faces challenges with dimensionality and overfitting, and LRL-Four's periodic nature limits its adaptability to non-periodic environments.

Table I outlines the mean values and 95% confidence bounds for training duration, operational cost, and error of various RL algorithms over a one-day operational simulation. Among the RL algorithms, LRL-TC stands out for its training efficiency, significantly outperforming LRL-Poly, LRL-Four, LRL-RBF, and DQN. Specifically, LRL-Poly, LRL-Four, LRL-RBF, and DQN required approximately 45.01, 22.45, 28.70, and 5.75 times longer to train than LRL-TC, respectively.

When examining operational costs, the algorithms were benchmarked against MILP outcomes. Here, DQN demonstrated remarkable precision with a minimal average error rate of just 0.78%. LRL-TC, with an average operational cost error of 2.92%, surpassed its counterparts, showcasing superior cost-effectiveness. LRL-RBF, despite a slightly higher error rate of 4.49%, still shows a promising inclination towards achieving optimal solutions but suffers from a training duration 3.89 times longer than DQN. Conversely, LRL-Poly and LRL-Four exhibited the highest errors at 12.70% and 12.74%.

B. Sensitivity Analysis of LRL-TC

As detailed in Section III-A, LRL-TC exhibits notable computational efficiency due to its binary structure and achieves performance levels close to DQN with fewer features over shorter training durations. A sensitivity analysis was conducted to determine whether LRL-TC can match or surpass DQN's accuracy while maintaining its computational advantages. This analysis involved exponentially increasing the tiles and tilings from 4 to 256 under consistent hyperparameter settings.

The operational costs associated with LRL-TC varied with changes in the number of tiles and tilings, as illustrated in Fig. 4. Initially, a decrease in costs was observed, dropping from 3234.26 to 3227.18 as the number of tiles and tilings increased from 4 to 16. A more substantial reduction occurred with the increment to 64, resulting in a cost of 3212.83. However, an uptick to 3233.06 was noted at 128 tiles and tilings, accompanied by higher variance due to stochastic influences on learning. Notably, at 256 tiles and tilings, operational costs significantly dropped to 3180.64, with 95% confidence bounds of (3171.98, 3189.31), closely approaching DQN's average operational cost of 3166.84 (3159.39, 3174.30). This suggests that with sufficient scaling, LRL-TC could potentially achieve parity with DQN's performance.

The training duration for LRL-TC increased exponentially with the number of features, defined by the power of the number of tiles and tilings and the input dimension. Initially, the durations were manageable, with averages of $107.42s$, $121.13s$, $142.03s$, and $185.65s$ for 4, 8, 16, and 32 tiles and tilings, respectively. However, a sharp rise in training time was observed from 64 tiles and tilings onward, recording $258.08s$, $428.39s$, and $749.59s$ for 64, 128, and 256 tiles and tilings, surpassing DQN's training duration of $618.62s$.

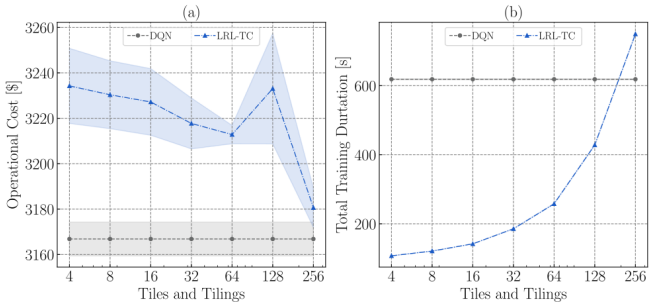

Fig. 4. Mean and 95% confidence bounds of (a) operational cost [$] and (b) total training duration [s] of LRL-TC and DQN.

C. Performance of LRL-TC and DQN on 3-Month Dataset

We utilized a diverse three-month dataset featuring varied operational scenarios to train and evaluate both the LRL-TC and DQN algorithms. To better accommodate the increased variation in the dataset, the ϵ decay rate was adjusted to 0.9995. As depicted in Fig. 5, both algorithms began their training with an average reward value of approximately -35 and showed improvement to -20 within the first 1000

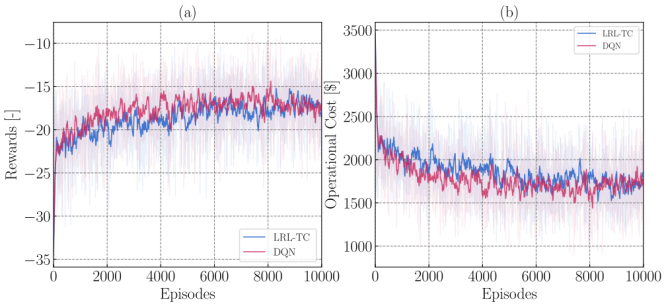

Fig. 5. The training results of LRL-TC and DQN on a three-month data set during 10000 episodes, average (a) rewards, (b) operation costs.

TABLE II
MEAN AND 95% CONFIDENCE BOUNDS OF LRL-TC'S AND DQN'S OPERATIONAL COST [$] ON TEST SET

Day	LRL-TC	DQN	MILP
1	424.56 (379.41, 469.72)	**358.26 (329.76, 386.76)**	146.37
2	2230.29 (2159.54, 2301.05)	**2137.56 (2096.12, 2179.00)**	1997.27
3	4572.42 (4526.40, 4618.45)	**4410.82 (4355.28, 4466.36)**	3988.14
4	1095.01 (1072.42, 1117.59)	**1025.64 (1001.02, 1050.25)**	838.38
5	1756.49 (1714.41, 1798.57)	**1641.84 (1619.30, 1664.38)**	1440.16
6	3717.35 (3680.74, 3753.97)	**3472.59 (3441.71, 3503.47)**	3358.59
7	2740.19 (2711.82, 2768.55)	**2607.74 (2573.39, 2642.09)**	2380.48
8	1974.95 (1922.82, 2027.07)	**1705.21 (1689.54, 1720.89)**	1464.88
9	2671.28 (2654.25, 2688.31)	**2549.12 (2515.98, 2582.26)**	2319.30
10	1761.44 (1749.45, 1773.42)	**1717.72 (1688.41, 1747.02)**	1616.36
11	1134.48 (1103.01, 1165.95)	**1114.39 (1071.85, 1156.94)**	884.00
12	5751.59 (5673.55, 5829.63)	**5649.82 (5592.84, 5706.79)**	5242.64
13	906.22 (886.95, 925.49)	**881.86 (836.93, 926.79)**	673.10
14	3569.42 (3537.85, 3600.99)	**3442.68 (3400.49, 3484.86)**	3142.40
15	1758.13 (1694.68, 1821.58)	**1668.28 (1636.15, 1700.40)**	1471.81

episodes. The rewards continued to gradually increase up to the 4000 episodes, followed by minor fluctuations until the end of the training period. Throughout the training, DQN consistently outperformed LRL-TC, achieving slightly higher rewards. The total training duration for LRL-TC and DQN was closely matched, with mean and 95% confidence bounds recorded at $762.46s(759.62, 765.29)$ for LRL-TC and $749.20s(747.68, 750.71)$ for DQN, indicating similar computational efficiency over the extended training period. The operational accuracy of the algorithms was further tested over 15 operation days, detailed in Table II. LRL-TC consistently underperformed relative to DQN. The average error percentage for LRL-TC was $30.63(28.10, 33.16)$, compared to $22.13(19.79, 24.48)$ for DQN. These results underscore a notable disparity in performance, with DQN demonstrating superior accuracy and generalization across diverse operational scenarios.

IV. CONCLUSION

This study explored the potential and limitations of using linear features in RL for the optimal energy dispatch of ESSs. Our investigation primarily focused on comparing the convergence, training efficiency, and operational performance of LRL strategies, specifically LRL-TC, against the more complex DQN algorithm, with MILP serving as the benchmark. Results indicate that LRL-TC is notably the most competitive among the LRL variants for its computation efficiency and operational cost accuracy. A sensitivity analysis was conducted by incrementally increasing the tiles and tilings, while LRL-TC demonstrates the potential to outperform DQN with a high number of tiles and tilings, enhancing its ability to handle environmental variations and noise. However, this benefit comes at the cost of significantly extended training durations. This trade-off highlights a crucial consideration for the application of LRL-TC in environments where training efficiency and operational performance are both priorities. Further extending the assessment to a three-month dataset revealed LRL-TC's challenges in generalization, demonstrating inconsistencies in adapting to varied and complex scenarios.

ACKNOWLEDGMENT

This work is supported by the Dutch National e-infrastructure SURF Cooperative (grant no. EINF-8477).

REFERENCES

[1] O. Ellabban, H. Abu-Rub, and F. Blaabjerg, "Renewable energy resources: Current status, future prospects and their enabling technology," *Renewable and sustainable energy reviews*, vol. 39, pp. 748–764, 2014.
[2] L. Zhang, Z. Yang, Q. Xiao, Y. Guo, Z. Ying, T. Hu, X. Xu, S. Khan, and K. Li, "Distributed scheduling for multi-energy synergy system considering renewable energy generations and plug-in electric vehicles: A level-based coupled optimization method," *Energy and AI*, vol. 16, p. 100340, 2024.
[3] M. Parol, T. Wójtowicz, K. Księżyk, C. Wenge, S. Balischewski, and B. Arendarski, "Optimum management of power and energy in low voltage microgrids using evolutionary algorithms and energy storage," *Int. J. of Electrical Power Energy Systems*, vol. 119, p. 105886, 2020.
[4] F. Garcia-Torres, C. Bordons, J. Tobajas, R. Real-Calvo, I. Santiago, and S. Grieu, "Stochastic optimization of microgrids with hybrid energy storage systems for grid flexibility services considering energy forecast uncertainties," *IEEE Trans. on Power Systems*, vol. 36, no. 6, pp. 5537–5547, 2021.
[5] Z. Lu, X. Xu, Z. Yan, and M. Shahidehpour, "Multistage robust optimization of routing and scheduling of mobile energy storage in coupled transportation and power distribution networks," *IEEE Trans. on Transportation Electrification*, vol. 8, no. 2, pp. 2583–2594, 2021.
[6] F. Han, J. Zeng, J. Lin, and C. Gao, "Multi-stage distributionally robust optimization for hybrid energy storage in regional integrated energy system considering robustness and nonanticipativity," *Energy*, vol. 277, p. 127729, 2023.
[7] D. Cao, W. Hu, J. Zhao, G. Zhang, B. Zhang, Z. Liu, Z. Chen, and F. Blaabjerg, "Reinforcement Learning and Its Applications in Modern Power and Energy Systems: A Review," *J. of Modern Power Systems and Clean Energy*, vol. 8, no. 6, pp. 1029–1042, 2020.
[8] D. Qiu, Y. Wang, W. Hua, and G. Strbac, "Reinforcement learning for electric vehicle applications in power systems:A critical review," *Renewable and Sustainable Energy Reviews*, vol. 173, p. 113052, 2023.
[9] Z. Zhang, D. Zhang, and R. C. Qiu, "Deep reinforcement learning for power system applications: An overview," *CSEE J. of Power and Energy Systems*, vol. 6, no. 1, pp. 213–225, 2020.
[10] X. Chen, G. Qu, Y. Tang, S. Low, and N. Li, "Reinforcement learning for selective key applications in power systems: Recent advances and future challenges," *IEEE Trans. on Smart Grid*, vol. 13, no. 4, pp. 2935–2958, 2022.
[11] R. S. Sutton and A. G. Barto, *Reinforcement Learning, second edition: An Introduction*. MIT Press, 2018.
[12] J. Cao, D. Harrold, Z. Fan, T. Morstyn, D. Healey, and K. Li, "Deep reinforcement learning-based energy storage arbitrage with accurate lithium-ion battery degradation model," *IEEE Trans. on Smart Grid*, vol. 11, no. 5, pp. 4513–4521, 2020.
[13] H. Shengren, E. M. Salazar, P. P. Vergara, and P. Palensky, "Performance comparison of deep rl algorithms for energy systems optimal scheduling," in *2022 IEEE PES Innovative Smart Grid Technologies Conference Europe*, 2022, pp. 1–6.
[14] E. M. Salazar Duque, J. S. Giraldo, P. P. Vergara, P. Nguyen, A. van der Molen, and H. Slootweg, "Community energy storage operation via reinforcement learning with eligibility traces," *Electric Power Systems Research*, vol. 212, p. 108515, 2022.
[15] H. Xu, A. D. Domínguez-García, and P. W. Sauer, "Optimal tap setting of voltage regulation transformers using batch reinforcement learning," *IEEE Trans. on Power Systems*, vol. 35, no. 3, pp. 1990–2001, 2020.
[16] S. Rammohan, S. Yu, B. He, E. Hsiung, E. Rosen, S. Tellex, and G. Konidaris, "Value-Based Reinforcement Learning for Continuous Control Robotic Manipulation in Multi-Task Sparse Reward Settings," 2021, arXiv:2107.13356.
[17] A. Rajeswaran, K. Lowrey, E. V. Todorov, and S. M. Kakade, "Towards Generalization and Simplicity in Continuous Control," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.
[18] S. Gao. [Online]. Available: https://github.com/ShuyiGao/LRLs