Symbolic Deep Reinforcement Learning for Energy Storage Systems Optimal Dispatch

Gao, Shuyi; Hou, Shengren; Palensky, Peter; Vergara, Pedro P.

DOI
10.1109/PowerTech59965.2025.11180187

Publication date
2025

Document Version
Final published version

Published in
2025 IEEE Kiel PowerTech, PowerTech 2025

Symbolic Deep Reinforcement Learning for Energy Storage Systems Optimal Dispatch

Shuyi Gao[1], Shengren Hou[1], Peter Palensky[1], Pedro P. Vergara[1]
[1]Intelligent Electrical Power Grids, Delft University of Technology, Delft, The Netherlands
s.gao@tudelft.nl, s.hou-1@tudelft.nl, p.palensky@tudelft.nl, p.p.vergarabarrios@tudelft.nl

2025 IEEE Kiel PowerTech | 979-8-3315-4397-6/25/$31.00 ©2025 IEEE | DOI: 10.1109/PowerTech59965.2025.11180187

***Abstract*—Reinforcement learning (RL) has become a promising approach for optimizing the dispatch of energy storage systems (ESSs) in distributed energy systems. Utilizing linear methods in the Q-representation of RL often struggles to balance accuracy and efficiency, while neural network (NN) performs well but falls short in terms of explainability and interpretability. To address these challenges, we developed and evaluated a deep-Q-symbolic-network (DQSN) framework, which integrates a symbolic network (SN) into the deep-Q-network (DQN) architecture for optimal dispatch of ESS. We benchmarked the performance of DQSN against DQN using mixed-integer linear programming (MILP) results, focusing on algorithm convergence, training duration, and operational cost accuracy. Our findings indicate that DQSN achieves slightly superior rewards and reduced operational costs with a modest increase in training time. Additionally, while DQN demonstrates superior generalization to unseen scenarios, DQSN excels in accurately fitting training data, enabling DQSN to be a viable alternative to DQN, particularly in applications requiring explainability and interpretability.**

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

NOTATION

Sets	
$\mathcal{B}$	set of ESSs
$\mathcal{V}$	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
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

I. INTRODUCTION

The integration of renewable energy sources (RES), such as photovoltaics (PVs) and wind turbines (WTs), increases complexity and volatility in distribution networks due to their intermittent nature, challenging the stable balance of supply and demand [1]. Energy storage systems (ESSs) improve energy supply consistency, reliability, and economic efficiency by storing excess energy during high renewable output or low electricity prices and releasing it during peak demand [2]. To solve the ESSs dispatch problem by modeling it into a mixed-integer linear program (MILP), model-based methods, such as stochastic and robust optimization, have been successful for ESSs dispatch but require precise operational knowledge and are computationally intensive [3]. In contrast, reinforcement learning (RL) transforms the dispatch problem into a Markov decision process (MDP), utilizing historical data to improve decisions without prior system models [4, 5].

RL finds optimal solutions by discretizing state and action spaces and updating a Q-function to evaluate the long-term returns of state-action pairs. To handle larger or continuous spaces, Q-functions are typically represented using parameterized forms such as linear functions (e.g., polynomials, Fourier series, radial basis functions, and tile coding) or a neural network (NN) [6]. The performance of using linear functions for updating the Q-function in ESS optimal dispatch has been extensively studied and compared in [7]. Although linear functions approach the operational cost accuracy of using neural networks, such as in deep-Q-network (DQN) [8], they face significant trade-offs in computational efficiency and struggle with generalization across larger and varied datasets. The work in [9] compared four advanced RL algorithms that utilized NNs for managing generators and ESSs. The results indicate that RLs leveraging NNs can deliver high-quality, real-time solutions even in unseen scenarios, demonstrating

the strong generalization capabilities of NNs and their ability to handle complex, high-dimensional nonlinear relationships.

Although NNs have achieved remarkable accuracy in representing Q-functions, they are seen as black-box models, providing limited insight into the learned mappings.

Symbolic Regression (SR) offers an explainable and interpretable alternative by searching for mathematical expressions that accurately and simply fit input and output data. Evolutionary methods (i.e., genetic programming) solve SR tasks by evolving candidate solutions through genetic operators such as crossover and mutation. However, these methods often increase complexity without performance gains [10]. Moreover, SR requires a population of formulas to evolve through generations, making it impractical for RL, where real-time interactions with the environment are essential. To address these problems, symbolic network (SN), an NN-structured SR approach, is introduced in [11]. SN represents the mappings using a heterogeneous NN with units implementing mathematical operators such as $+$, $-$, $\times$, $\div$, $sin(\cdot)$, $exp(\cdot)$, etc. SN's weights are adjusted using gradient-based methods to minimize training error, thus offering end-to-end training as an alternative to traditional NNs in RL. Furthermore, it reduces the number of active units in the network, enabling faster training than NN while maintaining its explainability and interoperability.

As ESSs play a crucial role in stabilizing electricity supply and optimizing energy usage, an efficient, accurate, and explainable method for optimal dispatch is required. Linear methods fail to balance accuracy and computational efficiency, while NN, although powerful, operates as a black-box model. SN meets all these requirements but has not yet drawn significant attention in this context. To address this gap and evaluate the potential of SN as a feasible alternative to NN, we developed a deep-Q-symbolic-network (DQSN) by embedding SN into DQN architecture, aiming to maintain the learning efficiency and generalization capability of DQN while maintaining its explainability. By employing MILP results as a benchmark, we conduct extensive simulations to evaluate DQSN's performance compared to traditional DQN, focusing on algorithm convergence, training duration, and operational cost accuracy. Notice that a bench of explanatory methods of SR-based models is conducted, such in [12]; in this paper, our focus lies on the performance of DQSN in ESS optimal dispatch.

II. Mathematical Model of ESSs Optimal Dispatch

A distribution system with ESSs, PVs, load demands, and an external power grid aims to minimize total operational costs over a day. This involves optimally scheduling ESS charging and discharging actions at each time step. The problem is a sequential decision-making challenge due to the time-dependent behavior of the ESS's state-of-charge (SOC), which is affected by previous charging/discharging operations. Additionally, decisions must account for time-varying uncertainties, including fluctuations in renewable energy generation, load demand variations, and dynamic electricity prices. The mathematical formulation of this optimal ESS dispatch problem is detailed 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) aims to minimize total operational costs by determining the ESS charge/discharge power $P^B_{i,t}$. The net load P^N_t is subject to (2) and constrained by the power import/export limits specified in (3). When $P^N_t > 0$, power is sold to the grid at the sell price $\sigma_t = \sigma_{s,t}$, and when $P^N_t < 0$, power is purchased from the grid at the buy price $\sigma_t = \sigma_{b,t}$. The maximum import/export capacities are represented by $\overline{P^G_t}$ and $\underline{P^G_t}$. Equations (4) and (5) define the ESS charge/discharge power and the SOC, respectively, where $P^B_{i,t} > 0$ indicates discharge and $P^B_{i,t} < 0$ indicates charge. The SOC evolution is modeled in (6), based on the defined charge/discharge actions.

III. MDP Formulation of ESSs Optimal Dispatch

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

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

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

At each time step, the state s_t comprehensively encapsulates the current operational dynamics of the distributed energy system, as defined in 7. The action space in (8) can be discretized by increments of charging/discharging power, such that each $P^B_{i,t} \in \underline{P^B_i}, \ldots, -2\Delta P^B_i, -\Delta P^B_i, 0, \Delta P^B_i, 2\Delta P^B_i, \ldots, \overline{P^B_i}$. Here, $\Delta P^B_i = \frac{1}{\mu_i - 1}(\overline{P^B_i} - \underline{P^B_i})$, with $\mu_i \in \mathcal{N}_+$. 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 ESS on idle mode [13].

The reward function guides RL algorithms in finding an optimal policy. It assigns a low value when agents make uneconomical or unsatisfactory decisions, indicated by the existence of ΔP_t in (9), where $\hat{P^N_t}$ represents the actual

power imported from the main grid. When $\hat{P_t^N}$ is positive, it opposes (10), and vice versa. In the formulated MDP, the reward function in (11) has two components that encourage the RL agent to consider both operational costs and power balance constraints. The weights φ_0 and φ_1 control the emphasis on these two objectives.

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

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

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

IV. Deep Q Symbolic Network

To solve the MDP, the DQN training focuses on learning an optimal Q-function $Q^*(s,a)$ that satisfies the Bellman optimality equation, thereby implicitly deriving the optimal policy. A policy network $Q(s,a;\omega)$ represents the parameterized Q-function and is updated to approximate the target Q-value by minimizing estimation errors. This involves adjusting the network's parameters to match the temporal-difference (TD) target, substituting for the unknown real value of $Q(s,a)$ The TD target $r + \gamma\hat{Q}(s',a';\hat{\omega})$ is generated by the target network $\hat{Q}(s,a;\hat{\omega})$. The target network is updated less frequently by periodically (every c episode) copying the parameters from the policy network with an update rate of τ, $\hat{\omega} \leftarrow \tau\omega + (1-\tau)\hat{\omega}$. The policy network is updated using gradient descent by sampling from the replay buffer $\mathcal{D} = \{(s,a,s',r) : s,s' \in \mathcal{S}, a \in \mathcal{A}\}$. The replay buffer stores and replays trajectories collected from interactions with the environment. By sampling mini-batches of experiences from the replay buffer, the network updates its parameters, which helps break the correlation between consecutive samples and leads to more stable and efficient learning [8]. The update of the policy network can be described as in (12).

$$\begin{aligned} \omega' &= \omega + \alpha(Q(s,a) - Q(s,a;\omega))\nabla Q(s,a;\omega) \\ &= \omega + \alpha(r + \gamma\hat{Q}(s',a';\hat{\omega})) - Q(s,a;\omega))\nabla Q(s,a,\omega) \end{aligned} \quad (12)$$

In this paper, based on the framework of DQN, a symbolic network [14] is adopted to represent the Q-function of state-action pairs within the continuous state space. SN integrates symbolic regression and a fully connected neural network structure, enabling an explainable and generalizable feature representation. In the symbolic network, as shown in Fig. 1, the usual choices (such as ReLU or tanh) in neural networks are replaced with mathematical manipulations. A L layer symbolic network can be described as:

$$\begin{aligned} y &= \mathbf{h}_{L+1} = \mathbf{W}_{L+1}\mathbf{h}_L \\ s.t. &\begin{cases} \mathbf{g}_n = \mathbf{W}_n\mathbf{h}_{n-1} \\ \mathbf{h}_n = f(\mathbf{g}_n) \end{cases} \end{aligned} \quad (13)$$

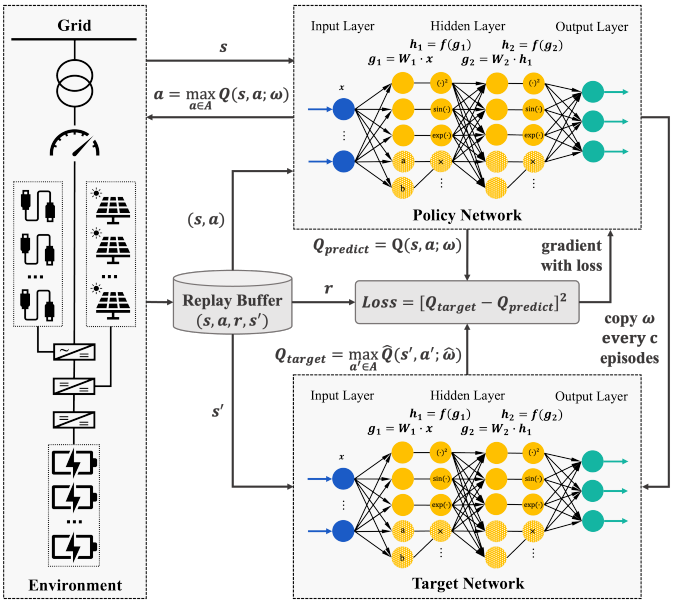

Fig. 1. Framework of DQSN in solving the ESS optimal energy dispatch.

The n-th layer comprises a weight matrix $\mathbf{W}_n$ and activation function $f(\mathbf{g}_n)$ with input data of each layer $\mathbf{h}_n$, and the input data of network $\mathbf{h}_0 = x$; the final layer does not have an activation function. The activation function consists of a separate function (e.g., $(\cdot)^2$, $sin(\cdot)$, $exp(\cdot)$ etc.) for each component or includes functions (e.g., $a \times b$.) that take two or more arguments while producing one output. The SN structure is powerful enough to represent complex environments and dynamics by stacking multiple layers, as the structure allows the fitting of complex combinations and compositions of various primitive functions. Additionally, it can be trained through backpropagation, enabling end-to-end training without multiple steps. More details about SN can be found in [14].

V. Case Study

A. Experimental Setting

The time interval between charge/discharge decisions is set to one hour, with each episode spanning $\mathcal{T} = 24$ hours. The selling price, $\sigma_{s,t}$, is set to $0.5\sigma_{b,t}$, where $\sigma_{b,t}$ represents the price of imported electricity from the main grid at time t. The grid's exchange capacity is established at $400kW$. The ESS parameters' capacity is set to $400kWh$, with the maximum and minimum charging/discharging power defined as $[-120, 120]$ kW. The SOC is constrained within $[0.2, 0.8]$, and the energy efficiency is considered 1. The action space is discretized into $\mu = 9$, and the initial SOC for each episode during training is set to 0.2. Parameters φ_0 and φ_1 are assigned values of 0.5 and 50, respectively. DQSN is implemented and compared with DQN to evaluate the effectiveness of SN in RL. Both NN and SN have three hidden layers, and the NN has 128 activate units in each layer, while the SN uses a set of operators $\{Constant(x)*2, Identity(x)*4, Square(x)*4, Sin(x)*2, Exp(x)*2, Sigmoid(x)*2, Product(x_1,x_2)*2\}$ in each layer. Following [14], a $L_{0.5}$ regularization is adopted

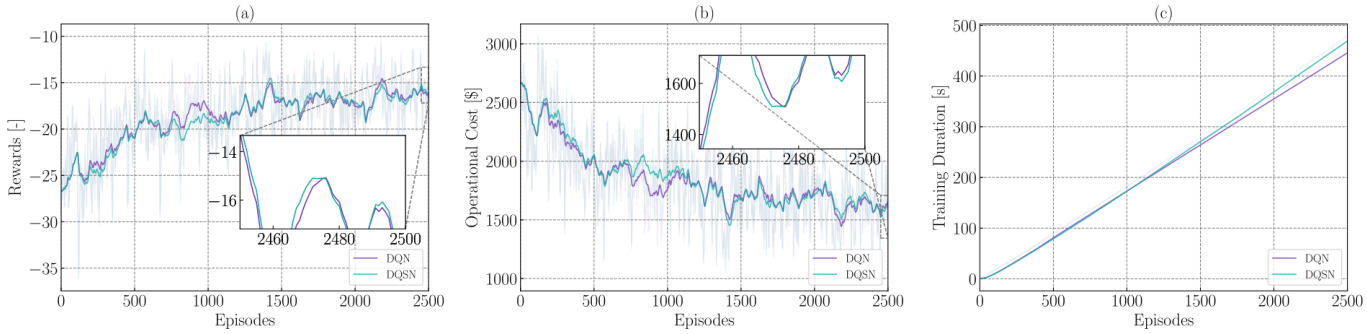

Fig. 2. The training of DQN and DQSN on a three-month data set during 2500 episodes, average (a) rewards, (b) operation costs, (c)training duration.

to enforce sparsity in the network, aiming to find the simplest possible equation that fits the data. Both algorithms are implemented in Python, and the MILP optimization problem is formulated and solved using the Pyomo package to provide benchmark results. The comparison focuses specifically on their convergence and training duration on the training dataset and optimization cost error on the test dataset. Operational cost error is determined by comparing the results of the DQN and DQSN algorithms against the MILP solution. Notice that the MILP solution is the global optimum as the model knows the future demand and generations for the whole decision period. Training duration refers to the computational time required for RL agents to converge, measured over 2500 episodes. Additionally, the average results from five random seed simulations are used to mitigate randomness. The implementation of DQSN is open-sourced in [15].

B. Performance on Training Set

Fig. 2 illustrates the average reward, operational cost, and training duration of DQN and DQSN over 2500 episodes. DQN and DQSN showed comparable rewards and operating cost patterns during the training process. Over the first 500 episodes, the rewards went from about -26 to -21 on average. Both algorithms demonstrated moderate gains, with rewards improving from around -21 to -17 over the next 1000 episodes. The rewards thereafter remained fairly consistent, rising from approximately -17 to -16. By the end of the training period, DQSN demonstrated a marginally higher reward compared to DQN. Operational costs displayed a near-symmetrical decreasing pattern relative to the rewards. However, between episodes 750 and 800, there was a slight deviation due to the imbalance induced by the non-optimal agent during training. Despite this, the overall pattern remained consistent by the end of training, reflecting the agents' successful adherence to power imbalance constraints. The training duration for both algorithms increased nearly linearly, with a total of 478.74 seconds for DQSN and 454.59 seconds for DQN. DQSN takes longer to train, despite having fewer activation units than DQN, due to the use of $L_{0.5}$ regularization. This regularization improves the learning of state-action features but compromises computational efficiency.

Overall, DQSN performs slightly better during training than DQN in terms of rewards and operational costs. Although requiring marginally more time to train, it presents as a competitive alternative to DQN that offers explainability and interpretability.

C. Performance on Test Set

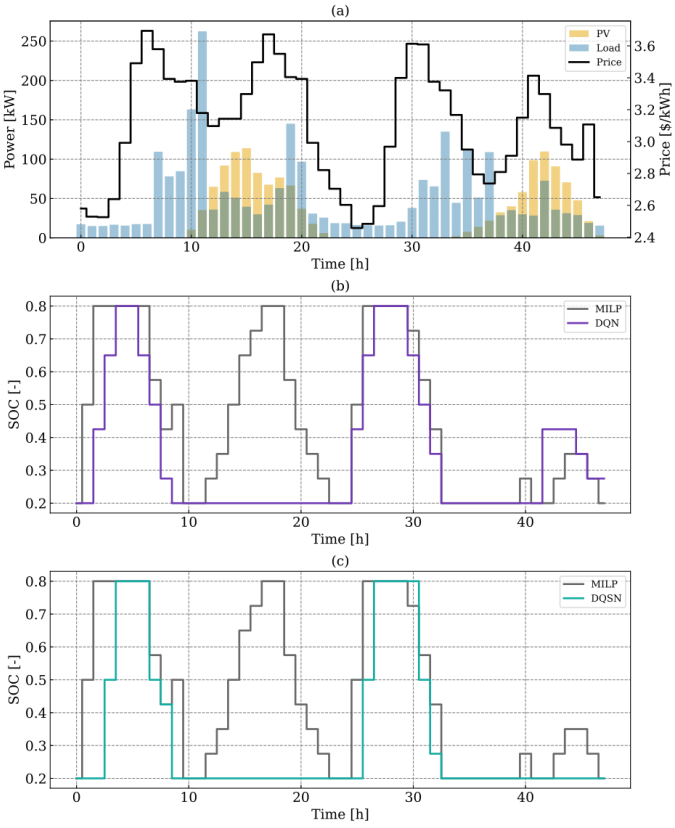

Fig. 3. Results of DQN and DQSN on two operational days, (a) PV generation, load consumption, and electricity price, (b), (c) comparison of ESS's SOC changing of the given charging and discharging operation strategy of DQN and DQSN.

Fig. 3(a) illustrates the dynamics of successive two test days in the test dataset, presenting PV generation, load demand, and electricity pricing, while Fig. 3(b) and Fig. 3(c) evolve the

TABLE I
MEAN AND 95% CONFIDENCE BOUNDS OF OPERATIONAL COST [$] OF DQN AND DQSN ON TEST SET

Day	MILP [$]	DQN Operational Cost[$]	DQSN Operational Cost[$]
1	1758.52	2090.79 (2036.01, 2145.57)	**2073.57 (1981.65, 2165.49)**
2	3793.30	4493.24 (4437.52, 4548.95)	**4475.82 (4422.28, 4529.35)**
3	765.04	**982.69 (961.15, 1004.23)**	1067.17 (997.49, 1136.84)
4	1338.70	**1645.26 (1609.36, 1681.15)**	1737.59 (1696.09, 1779.09)
5	3233.31	**3529.67 (3498.73, 3560.60)**	3550.55 (3516.83, 3584.28)
6	2133.76	**2690.04 (2645.90, 2734.18)**	2717.82 (2661.23, 2774.42)
7	1288.45	1831.06 (1803.23, 1858.90)	**1743.01 (1711.46, 1774.56)**
8	2150.62	2655.02 (2576.39, 2733.66)	**2649.49 (2571.46, 2727.52)**
9	1516.37	**1715.57 (1699.49, 1731.65)**	1753.72 (1718.26, 1789.19)
10	694.33	**1045.74 (994.94, 1096.54)**	1092.03 (1030.35, 1153.70)
11	4861.71	**5623.26 (5530.24, 5716.29)**	5640.66 (5598.82, 5682.51)
12	551.72	**822.67 (779.95, 865.39)**	851.76 (802.60, 900.93)
13	2960.35	3523.68 (3383.79, 3663.56)	**3451.34 (3416.35, 3486.33)**
14	1369.99	**1631.91 (1565.83, 1697.99)**	1794.24 (1720.05, 1868.44)

SOC of the ESS based on the learned charging/discharging strategies of DQN and DQSN, alongside the global optimal solution derived from MILP.

The optimal strategy derived from MILP starts charging at a low price (e.g., time step 1, 11, 25) or high PV generation (e.g., time step 15, 42) and discharging while experiencing high load demand (e.g., time step 10, 18) or high price (e.g., time step 6, 30), and the total operational cost reach at 3666.99 [$]. DQN closely aligns with the optimal charging and discharging strategy except for the period between 11 and 23, achieving a cost of 4680.48 [$]. Conversely, DQSN correctly identifies the first price peak but misses subsequent charging and discharging opportunities on both days. Despite this, as the second price peak comes with high PV generations, which compensate for the cost, DQSN surprisingly incurs a lower cost than DQN, amounting to 4117.26 [$].

Table I outlines the mean values and 95% confidence bounds for operational cost and its error against MILP results over 14 days of operational simulation in the test dataset are shown in Fig.4. For average operational costs, out of the 14 days, DQN demonstrates higher accuracy on 9 days, whereas DQSN outperforms DQN on 6 days. Regarding the operational cost error, DQSN demonstrates higher variance than DQN on most of the test on days 1, 3, 7, 8, 9, and 12 while lower variance on days 2, 4, 5, 6, 10, 11, 13, and 14. The average errors for DQN and DQSN are 25.44% and 27.98%, respectively.

While Section.V-B and the case depicted in Fig.3 show DQSN achieved higher rewards and lower operational costs than DQN, the performance with multiple randomness (as in Table I and Fig.4) indicates that DQN's average performance is marginally superior to DQSN. This suggests that DQN can better generalize to unseen scenarios, whereas DQSN is more adept at fitting the relationships between state-action pairs and their Q-value within the training data.

VI. CONCLUSION

In this paper, we presented the development and evaluation of the DQSN, which integrates SN into the DQN architecture. Employing results from MILP as benchmarks, we conducted extensive simulations to compare the performance of DQSN and DQN in terms of algorithm convergence, training duration, and operational cost accuracy. DQSN achieves slightly higher rewards and operational costs during training, although with increased training time, which is an acceptable trade-off. This indicates that DQSN is a viable substitute for DQN since it offers competitive performances and provides interpretability and explainability. Furthermore, while DQN demonstrates superior generalization to unseen scenarios, DQSN's strength lies in its accurate fitting of training data, positioning it as a competitive option for applications where explainability and interpretability are crucial.

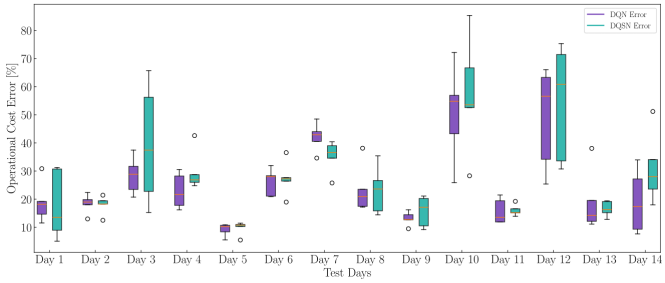

Fig. 4. Operation cost error of DQN and DQSN on the test set.

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] 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.

[4] 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.

[5] 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.

[6] R. S. Sutton and A. G. Barto, *Reinforcement Learning, second edition: An Introduction*. MIT Press, 2018.

[7] S. Gao, S. Hou, E. M. S. Duque, P. Palensky, and P. P. Vergara, "Linear reinforcement learning for energy storage systems optimal dispatch," in *2024 IEEE PES Innovative Smart Grid Technologies Europe*. IEEE, 2024, pp. 1–6.

[8] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.

[9] 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.

[10] J. Kubalík, E. Derner, and R. Babuška, "Toward physically plausible data-driven models: A novel neural network approach to symbolic regression," *IEEE Access*, 2023.

[11] G. Martius and C. H. Lampert, "Extrapolation and learning equations," *arXiv preprint arXiv:1610.02995*, 2016.

[12] N. Makke and S. Chawla, "Interpretable scientific discovery with symbolic regression: a review," *Artificial Intelligence Review*, vol. 57, no. 1, p. 2, 2024.

[13] 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.

[14] S. Kim, P. Y. Lu, S. Mukherjee, M. Gilbert, L. Jing, V. Čeperić, and M. Soljačić, "Integration of neural network-based symbolic regression in deep learning for scientific discovery," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 9, pp. 4166–4177, 2021.

[15] S. Gao. [Online]. Available: https://github.com/ShuyiGao/RL_SR