Comparative Analysis of Reinforcement Learning for Carbon-Aware Energy Management Systems

WeiChun Chang, Young-ho Cho, and Yeji Kim

The University of Texas at Austin, Austin, TX
{weichun.chang, jacobcho, Yeji Kim}@utexas.edu

***Abstract*—This paper investigates the application of reinforcement learning (RL) for optimal energy management in modern infrastructures, focusing on buildings. The vast time-series data generated by sophisticated control systems presents an opportunity to optimize battery charging and scheduling in real-time. This approach addresses the key challenge of balancing fluctuating energy demands with available resources, including renewable, to achieve economic and environmental benefits. The proposed carbon-aware EMS can learn and adapt to dynamic energy demands, variable renewable energy production, fluctuating electricity prices, and carbon emissions. This allows them to optimize for multiple objectives simultaneously, minimizing energy costs while meeting demand constraints, respecting battery capacity limitations, and incorporating carbon pricing to minimize pollution.**

***Index Terms* – Energy Management Systems, Reinforcement Learning, Carbon Emission Reduction, Carbon Tax**

Key Contributions

- We propose an RL-based model that leverages time-series data for precise real-time prediction, enabling the creation of an optimal charging and discharging schedule for an Energy Management System (EMS).
- The model considers various factors, including electricity prices, load demand, and renewable energy availability, to harmonize supply and demand while minimizing carbon emissions. We address the intricate nature of the problem, where parameters like tariffs and solar power are dynamic, and constraints exist due to battery limitations and charging rates. This work highlights the potential of RL methods to effectively inform decision-making for energy management systems.

I. Introduction

Modern infrastructures, such as buildings, rely on advanced control systems that produce extensive time-series data. This data captures real-time energy consumption patterns and the availability of energy sources, including renewable. By effectively analyzing this data with advanced control strategies, we can optimize battery charging for buildings efficiently.

A. Challenges and Existing Solutions

The primary challenge lies in achieving an optimal balance between energy demand and supply for cost-effective solutions. While various approaches, including deep learning, machine learning, and AI, have been proposed for battery management, existing solutions often focus on optimizing individual aspects of the system, such as charging scheduling or battery management. The existing solutions are listed below:

- Research has explored formulating charging schedules as non-convex optimization problems [1]
- Model predictive control (MPC) has been applied to optimize battery management systems (BMS), considering both stochastic and robust approaches [2], [3]
- Research on nonlinear MPC explores optimizing battery charging itself, with some studies focusing on time-series data [4], [5], [6].
- Integrating battery energy storage (BES) requires bridging the gap between detailed low-level battery charging constraints and high-level battery operation models. Research like [7] addresses this challenge.

B. Incorporating Renewables

The increased integration of renewable energy sources, while environmentally beneficial, poses challenges due to their inherent variability. Energy management systems (EMS) need to account for this instability when controlling charging and discharging patterns. Research has addressed this by a general Electric Vehicle-Intelligent Energy Management and Charging's Scheduling System [8]. In addition, traditionally, pricing models focused solely on minimizing charging costs, including electricity demand and energy charges. However, with the goal of achieving net-zero emissions, recent studies incorporate carbon pricing into the optimization process. This approach encourages the utilization of renewable energy sources in a practical way. Research such as [9] highlights the relationship between the energy sector, economic activity, and carbon pricing. Such models can be valuable tools for monitoring carbon price dynamics.

C. Optimizing EMS for Economics Efficiency and Sustainability with RL

Traditionally, pricing models focused solely on minimizing charging costs, including electricity demand and energy charges. However, with the goal of achieving net-zero emissions, recent studies incorporate carbon pricing into the optimization process. This approach encourages the utilization of renewable energy sources. Overall, traditional control strategies often struggle with the complexities of modern energy management systems, particularly in time-dependent settings. Reinforcement learning (RL) offers a compelling alternative. Here's how RL addresses the key challenges we've outlined:

1) Dynamic and Uncertain Environment: Unlike pre-programmed models, RL can learn and adapt to the real-time dynamics of the system. This includes fluctuating energy demands, variable renewable energy production, and evolving electricity prices.

2) Multi-Objective Optimization: RL excels at balancing multiple objectives simultaneously. It can optimize for minimizing energy costs while meeting energy demands, respecting battery capacity limitations, and incorporating carbon pricing to minimize pollution.

3) High-Dimensional Decision Space: Modern energy management systems involve a vast number of variables and complex interdependencies. RL can effectively navigate this high-dimensional decision space to find optimal solutions.

In essence, the proposed RL based model acts as a self-learning agent that interacts with the real-time environment, receives feedback on its actions (energy cost, battery health, etc.), and continuously refines its decision-making strategy to achieve the defined goals.

Existing research has touched upon RL in various control domains [10]. However, within the specific contexts of battery charging for buildings [11], comprehensive comparative studies between RL are relatively scarce. This work aims to bridge this gap by providing a detailed comparison between these control methodologies in energy management applications.

II. System Modeling for Energy Management System

We consider the dynamics within an Energy Management System (EMS) as captured in Fig. 1, which articulates the transactional flow between photovoltaic (PV) power generation P_t, building energy demand L_t, grid electrical supply G_t, and an energy storage system (ESS). At each decision epoch t, the PV installation has the potential to fulfill immediate building energy requirements P_t^L or to charge the energy storage unit P_t^E, depending on the surplus energy available.

The building's energy needs at any instance t are met through a synthesis of direct PV productionP_t^L, the battery's energy release E_t^L, or power from the grid G_t^L, with the decision driven by the gap between local generation and consumption. The grid stands as a responsive participant in this energy ecosystem, offering power as needed for building consumptionG_t^L and battery charging G_t^E while also purchasing excess energy from the PV system P_t^G.

Central to this operation is the battery, tasked with the dual role of stabilizing the energy supply through strategic charging and discharging actions. The energy flow from and to the battery is precisely controlled, factoring in the grid's variable electricity pricing λ_t^E, which dictates the economic viability of each transaction.

In this system, the EMS is designed to achieve an optimal balance, ensuring energy demands are met cost-effectively and sustainably. It leverages the variable output of the PV system, the flexibility of the battery, and the grid's ability to smooth supply and demand discrepancies. The goal is to align financial and environmental objectives, enhancing renewable use and reducing emissions. The energy management strategy is structured around a decision-making model, which will be detailed through an MDP framework focusing on transition dynamics and reward mechanisms.

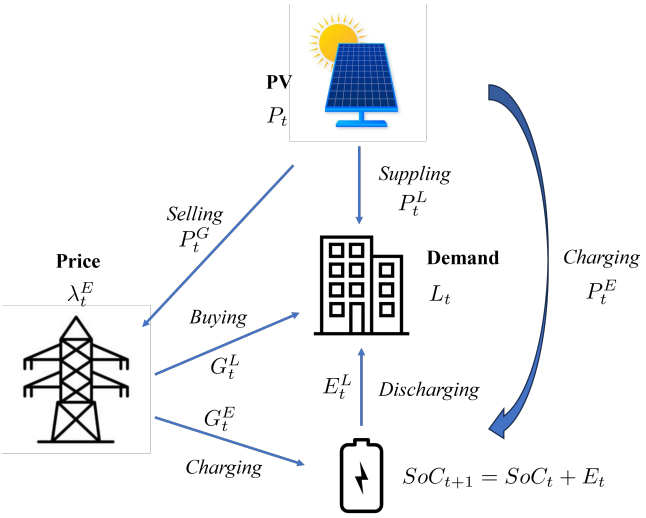

Fig. 1: **EMS:** An EMS operator navigates the intricate interplay between the grid, building, photovoltaic (PV), and battery within a finite horizon T, striving to craft an optimal strategy. The overarching goal centers on orchestrating these elements to maximize efficiency and balance energy utilization within the prescribed horizon.

Markov Decision Process (MDP) on Energy Management System

The application of reinforcement learning to the moment-by-moment battery management system involves the utilization of a Markov Decision Process (MDP) to facilitate structured decision-making. In this framework, the system aims to optimize its purchasing scheme for electricity, considering variant factors such as load demand, energy price, solar power generation, and the state of charge (SoC) of the battery, all of which evolve over time. Here we define four components of our MDP modeling system, including set of states, set of actions, transition dynamics, and reward functions.

Set of States ($\mathcal{S}$):

$$\mathcal{S} = \{s_t \mid s_t = (L_t, P_t, \text{SoC}_t, \lambda_t^E)\}$$

L_t : electricity load demand at time t (kW)
P_t : PV generation at time t (kW)
SoC_t : state of charge of the battery at time t
λ_t^E : price of grid electricity at time t ($/kWh)

Set of Actions ($\mathcal{A}$):

$$\mathcal{A} = \{a_t \mid a_t = (E_t, E_t^L, P_t^L, P_t^E)\}$$

E_t : the battery charge/discharge at time t

E_t^L : Energy from the ESS to the load at time t

P_t^E : Energy from the PV to the battery at time t

P_t^L : Energy from the PV to the load at time t

The set of actions in this MDP context reflects the control over the battery's charging or discharging behavior. The action E_t can be continuous or discrete, and it includes:

- Charging ($E_t > 0$): Indicates the process of replenishing the battery's energy by drawing electricity from external sources.
- Discharging ($E_t < 0$): Involves utilizing the energy stored in the battery to meet electricity demand or feed excess energy back into the grid.
- Neutral state ($E_t = 0$): Maintains the battery in a state of neither charging nor discharging.

Additionally, the energy supplied by the grid (G_t) dynamically adjusts based on the electricity demand (L_t) and the chosen action (E_t), ensuring efficient utilization of available resources while meeting operational requirements. In other words, any energy shortfall not covered by the battery is compensated for by the grid. To compute the cost of energy from grid, we are interested in the following decision variables.

1) Energy Supplied by the Grid: Assuming that every energy not met by the battery is met by the grid; the following equality holds for charge and discharge.

$$G_t = L_t + E_t - P_t$$

2) Decision Variables: Based on the actions taken by the battery, we can determine other decision variables. These decision variables include:

$$\begin{aligned} G_t &= G_t^L + G_t^E - P_t^G \\ L_t &= G_t^L + E_t^L + P_t^L \\ P_t &= P_t^G + P_t^E + P_t^L \\ E_t &= G_t^E + P_t^E - E_t^L \end{aligned}$$

Where G_t^L, G_t^E, P_t^G, E_t^L, P_t^L, and P_t^E represent the respective components of the energy supplied by the grid, energy stored in the battery, grid power, energy lost during charging, power consumed by loads, and power expended during discharging.

3) Carbon Emission: Finally, the carbon emissions at time t (C_t) are calculated to consider the environmental impact of the battery management system's operations. This calculation incorporates the carbon emissions associated with grid electricity and energy exchange actions:

$$C_t = w^G G_t + w^E |E_t|$$

Where:

- w^G represents the weight associated with the carbon emissions of grid electricity (kg CO_2/kWh) .
- w^E represents the weight associated with the carbon emissions of energy exchange actions (charging or discharging (kg CO_2/kWh)).

Transition Dynamics

The state of charge (SoC), denoted as SoC_t, represents the amount of energy stored in the battery entity (e.g., car or building) at time t. It's worth noting that SoC has a specific capacity. Transition dynamics describe how the battery's SoC evolves from one time step (t) to the next ($t+1$) based on the current SoC (SoC_t) and the chosen action (E_t). It ensures that the SoC remains within specified bounds ($\text{SoC}_{\min}$ and $\text{SoC}_{\max}$). The transition dynamics for the SoC of the battery from time t to $t+1$ are defined as follows:

$$\text{SoC}_{t+1} = \begin{cases} \text{SoC}_{\min} & \text{if } \text{SoC}_t + E_t < \text{SoC}_{\min} \\ \text{SoC}_{\max} & \text{if } \text{SoC}_t + E_t > \text{SoC}_{\max} \\ \text{SoC}_t + E_t & \text{otherwise} \end{cases} \quad (1)$$

Reward Function

The reward function (r_t) quantifies the immediate benefit at time t by incorporating the grid electricity price (λ_t^E), the energy supplied by the grid (G_t), and the and the cost of carbon emissions (C_t). It is formulated as:

$$r_t = \lambda_t^E \cdot G_t + \lambda^C \cdot C_t$$

This equation provides a straightforward representation of the reward obtained at each time step based on these key factors. Note that we only utilize energy costs as a reward function for the basic EMS model.

Discount Factor

The proposed problem is an episodic task. We set the discount factor as $\gamma = 1$, indicating that future rewards are fully considered in the agent's decision-making process.

III. Reinforcement Learning

In this section, we provide an overview of prominent RL algorithms, highlighting their distinct advantages and disadvantages. The choice of an algorithm depends on the specific characteristics of the environment and the objectives of the learning task.

Four different reinforcement learning algorithms have been implemented.

A. Advantage Actor-Critic (A2C)

Advantage Actor-Critic (A2C) [12] is a reinforcement learning architecture where the policy network (actor) proposes actions and the value network (critic) evaluates them. It utilizes the advantage function to measure the potential reward of a specific action relative to the average reward for the current state, enhancing the actor's ability to make better-informed decisions. This setup encourages the agent to prioritize actions that could yield higher rewards than the norm, thus driving

more effective exploration and exploitation of the learning environment.

A2C improves upon the basic actor-critic method by collecting experience in parallel across multiple agents and updating policies synchronously, which reduces the update variance and increases the stability of the learning process. This parallelization allows A2C to leverage computational resources efficiently, facilitating faster learning across diverse scenarios. However, A2C can struggle with environments that offer sparse rewards, rendering it sub-optimal [13]. In such cases, its inherent exploration strategies might not adequately sample the environment, leading to increased learning variance and slower convergence rates. Effective tuning of hyperparameters, such as the discount factor and learning rates, alongside techniques to encourage exploration, like adding noise to actions or varying policy parameters, can help mitigate these issues and optimize the algorithm's performance in challenging environments.

B. Trust Region Policy Optimization (TRPO)

Trust Region Policy Optimization (TRPO) [14] adopts a meticulous approach to updating policies by ensuring minimal deviation from established strategies, thereby maintaining a stable training environment and averting drastic performance declines. This stability is achieved by setting a constraint on the KL divergence between the old and new policies—referred to as the "trust region." Such constraints ensure that, although the policy is being refined iteratively, the updates remain small and manageable, reducing the risk of destabilizing the learning process.

TRPO is particularly favored in scenarios involving complex action spaces and environments where precise control over policy updates is crucial. It achieves robust performance improvements by carefully managing how the policy evolves, ensuring that each step is a controlled, incremental change rather than a radical shift. However, the reliance on second-order optimization methods makes TRPO computationally intensive, involving calculations that can be costly in terms of time and resources. This complexity not only makes it challenging to implement but also limits its application in environments where quick decisions are crucial or where computational resources are limited. As such, while TRPO is highly effective in ensuring safe and stable policy improvement, its practical deployment needs to be carefully considered in the context of available computational capabilities and the specific requirements of the task at hand.

C. Proximal Policy Optimization (PPO)

Proximal Policy Optimization (PPO) [15] aims to simplify the implementation of policy gradient methods while retaining similar benefits as more complex algorithms like TRPO. PPO introduces a clipped surrogate objective function, which puts a boundary on how much the policy can change in a single update, thus preventing harmful large updates. This clipping mechanism is a key feature that allows PPO to be optimized using first-order methods, significantly reducing the computational burden compared to second-order methods used in TRPO.

PPO has gained popularity due to its effectiveness and simplicity, especially in environments with large and continuous action spaces. The clipping parameter in PPO plays a critical role in maintaining the balance between sufficient exploration and the stability of the learning process. However, finding the right setting for the clipping parameter can be challenging as it needs to be tuned according to the specific characteristics of the environment. Moreover, PPO can struggle in scenarios with discrete action spaces or sparse rewards, where the algorithm might prematurely converge to suboptimal policies [16]. In such cases, additional modifications and tuning of hyperparameters are required to ensure that PPO can still deliver robust performance across a wide range of reinforcement learning tasks.

D. Recurrent Proximal Policy Optimization (Recurrent PPO)

Recurrent Proximal Policy Optimization (Recurrent PPO) [17] builds on the strengths of PPO by integrating recurrent neural networks (RNNs) into the framework, enhancing its capability to handle environments with strong temporal dependencies. By incorporating a hidden state that acts as a memory component, Recurrent PPO can remember and utilize past information to make more informed decisions. This feature is particularly valuable in tasks where the agent's current decision depends significantly on its previous actions or the historical state of the environment.

The addition of RNNs allows Recurrent PPO to excel in complex scenarios such as partially observable environments or tasks that require maintaining information over long time horizons. However, the integration of recurrent architectures adds layers of complexity to the training process. Managing sequence lengths and ensuring efficient training without overfitting or underfitting requires careful architectural and hyperparameter adjustments. Additionally, the computational demand of training RNNs can be substantial, necessitating significant computational resources, which may limit the scalability of Recurrent PPO in resource-constrained settings.

IV. Evaluation

By employing optimal control algorithms, we addressed challenges in Energy Management Systems (EMS), evaluating their comparative performance. Reinforcement Learning (RL) training occurred on a laptop housing an Intel® CPU @ 2.30 GHz, 16 GB RAM, and Intel® UHD 620 GPU (8GB VRAM). The baseline model was formulated using Pyomo [18], and the RL algorithms were implemented via Stable-Baseline3 [19]. This approach facilitates a streamlined evaluation of algorithmic efficacy across EMS context.

EMS Domain

The study conducts a comparative analysis of Reinforcement Learning (RL) algorithms within the domain of EMS. Utilizing hourly demand data and PV generation from a commercial building in Korea during July 2017 and Real-Time (RT) prices from Pennsylvania, New Jersey, and Maryland (PJM) region - chosen for their similar latitudes to Korea's market structure - the research aims to evaluate algorithmic performances.

Key parameters and constraints were established: State of Charge (SoC) range for battery was normalized from zero to one, with operational limits set at SoC values of 0.2 and 0.8. Battery charge/discharge limits were defined with $E_{\max}$ set as 0.1. For carbon emission calculation and costs, we set λ^C as 25$/Tons of CO2 and $\{w^G, w^E\}$ as $\{0.202, 0.083\}$kg CO2/kWh. During the RL algorithms training, we also include month and weekday/weekend information as state $\mathcal{S}$ to improve the training efficiency. We discretize E_t as $\{-0.1, -0.05, 0, 0.05, 0.1\}$ and E_t^L as $\{0, 0.05, 0.1\}$. For distributing the PV generation, we discretize P_t^L and P_t^E as $\{0, 0.05, 0.1\}$ to avoid exceeding the battery charging/discharging rate.

1) Baseline: In general, optimization-based control algorithms are not considered as a baseline since solving optimization problems at each time step is not practical for large-scale systems such as power systems and EMS. However, we need to set up the baseline model to discuss the effectiveness of the proposed EMS. To tackle this issue, we set the baseline model that only utilizes the PV generators. Since the model does not use battery, the PV generations are fully supplied to the load demand $P_t = P_t^L$. Then, the energy buying from the grid G_t can be easily calculated by the difference between the load demand and the PV generation as $L_t - P_t$.

2) RL: We propose two types of RL models, which are EMS with/without considering carbon emission costs. The parameters for the reinforcement learning algorithms are found in Table I. Fig. 2 illustrates the learning curves of EMS for four distinct algorithms. We utilize four RL algorithms that Stable-Baseline3 supports for multi-discrete action spaces: TRPO, PPO, Recurrent PPO, and A2C. Notably, TRPO and PPO exhibit commendable convergence with a notable degree of smoothness and stability in their learning processes. Compared to the two algorithms, recurrent PPO fails to learn the claimed performance and has the highest computational burden. The challenges faced by A2C in achieving effective learning may find potential mitigation by replacing the feed-forward neural network policy with a recurrent policy. Even though A2C has the fastest convergence in the training phase, it cannot reach the optimal point. Fig. 3 illustrates the learning curves of carbon-aware EMS for four distinct algorithms. By considering the carbon emission costs, the total costs of carbon-aware EMS are greater than those of the basic EMS. Similar to basic EMS training results, TRPO and PPO also exhibit commendable convergence with a notable degree of smoothness and stability in their learning processes. Conversely, Recurrent PPO and A2C cannot reach the optimal point.

TABLE I: Parameters of the RL Models for BMS

Parameters	Value
Total timesteps	2×10^6
Steps per episode	720 (1 month)
Number of parallel environments	8

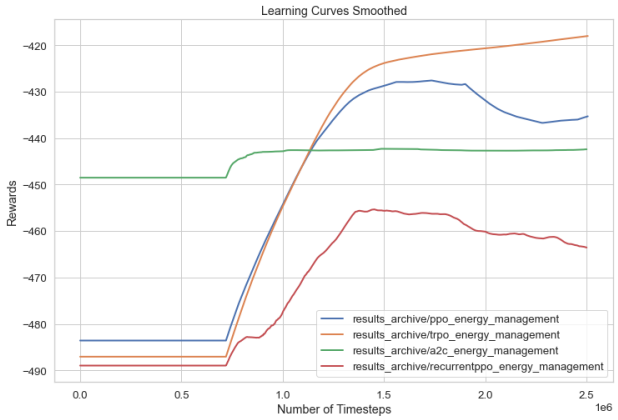

Fig. 2: Learning Curves of basic EMS for Various RL Algorithms such as PPO (blue), TRPO (orange), Recurrent PPO (red), and A2C (green).

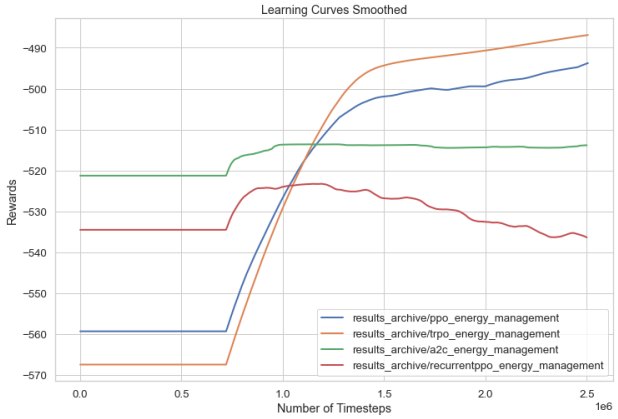

Fig. 3: Learning Curves of carbon-aware EMS for Various RL Algorithms such as PPO (blue), TRPO (orange), Recurrent PPO (red), and A2C (green).

3) Comparison between the baseline model and EMS with RL algorithms: We compare the control results of the baseline and EMS with the proposed carbon-aware EMS. The two foremost RL models, TRPO and PPO, are juxtaposed with the baseline. A test dataset encompassing a month's worth of data has been utilized, ensuring that the basic and carbon-aware EMSs have had no prior exposure to this dataset.

We first compare the energy costs of the basic EMS, carbon-aware EMS, and the baseline model. Results demonstrate that RL methods surpassed the baseline model, exhibiting a notable 40% enhancement in energy cost efficiency, equating to approximately $7,000 in monthly savings. Both PPO and TRPO-based EMSs significantly reduced energy costs. Since carbon-aware EMSs consider carbon emission costs to be their reward function, the energy costs of carbon-aware EMSs are slightly higher than those of basic EMSs.

We also compare the carbon costs of the basic EMS, carbon-aware EMS, and the baseline model. Notably, carbon-aware EMS has a lower cost than basic EMS and the baseline. The basic EMS only tries to minimize the energy cost, which inevitably leads to a higher carbon cost. The baseline model

does not control the battery, leading to a higher carbon cost.

We should highlight one of the shortages of the proposed EMS. Both types of EMS need a sufficient number of training data samples. This outcome can also explain why the recurrent PPO has a higher cost than other RL algorithms and cannot train the optimal policy. In the case of power system problems, obtaining a sufficient amount of data without any measurement noise is quite hard. Thus, the lack of training data may have led to incorrect interpretation of the simulation results. In future work, the consideration of measurement noise and uncertainties can be included for the MDP of EMS design.

Nonetheless, the carbon-aware EMSs with RL algorithms are well-trained based on the MDP setup while maintaining the sustainable computation burden. By considering the advantage of RL algorithms for online decision-making procedures, the proposed carbon-aware EMS can be applied to the real-time operation of building management to reduce both energy and carbon costs.

TABLE II: Comparison of EMS Performance between RLs and the baseline model.

Controller	Energy Cost ($)	Carbon Cost ($)	Total ($)
Baseline	16,971.7	2,981.63	19,953.33
PPO	10,560.65	2,540.17	13,100.82
TRPO	**10,080.72**	2,500.39	12,581.11
PPO with carbon	11,290.67	1,750.11	13,040.78
TRPO with carbon	10,700.42	**1,680.13**	**12,380.55**

V. Conclusion

This study presents an in-depth comparative analysis of reinforcement learning (RL) algorithms for Energy Management Systems (EMS), with a specific focus on natural policy gradient methods like Proximal Policy Optimization (PPO) and Trust Region Policy Optimization (TRPO). Our experimental results demonstrate a significant enhancement in energy cost efficiency, with these methods achieving over a 40% reduction, which translates to approximately $7,000 in monthly savings. Importantly, the inclusion of carbon cost considerations in carbon-aware EMS not only further reduces energy costs but also lowers carbon emissions effectively.

The practical application of these findings is evidenced by the successful deployment of carbon-aware EMS in real-time operations, which supports the integration of battery storage and renewable energy sources. This capability significantly contributes to the operational efficiency and sustainability of energy systems.

Moving forward, exploring Transfer Reinforcement Learning, as outlined in prior work [20], holds potential for enhancing the scalability of our approach. This strategy could enable the efficient transfer of learned policies across diverse EMS settings, including different types of residential, commercial, and industrial buildings, thereby accelerating the training processes and improving system adaptability.

In summary, the application of advanced RL techniques, particularly PPO and TRPO, has proven highly effective in optimizing energy management for cost and carbon efficiency. These techniques not only support the immediate financial benefits but also contribute to broader environmental goals by minimizing carbon emissions. The robust performance of these RL approaches in real-world settings underscores their potential to transform energy management practices, making them a vital tool for achieving sustainable energy solutions.

References

[1] Y. He, Z. Liu, and Z. Song, "Optimal charging scheduling and management for a fast-charging battery electric bus system," *Transportation Research Part E: Logistics and Transportation Review*, vol. 142, p. 102056, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1366554520307079

[2] A. Pozzi and D. M. Raimondo, "Stochastic model predictive control for optimal charging of electric vehicles battery packs," *Journal of Energy Storage*, vol. 55, p. 105332, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352152X22013287

[3] F. Jiao, Y. Zou, X. Zhang, and B. Zhang, "Online optimal dispatch based on combined robust and stochastic model predictive control for a microgrid including ev charging station," *Energy*, vol. 247, p. 123220, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360544222001232

[4] Y. Wang, C. Zhou, and Z. Chen, "Optimization of battery charging strategy based on nonlinear model predictive control," *Energy*, vol. 241, p. 122877, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360544221031261

[5] A. Di Giorgio, F. Liberati, and S. Canale, "Electric vehicles charging control in a smart grid: A model predictive control approach," *Control Engineering Practice*, vol. 22, pp. 147–162, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0967066113001871

[6] N. Tian, H. Fang, and Y. Wang, "Real-time optimal lithium-ion battery charging based on explicit model predictive control," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 1318–1330, 2021.

[7] H. Pandžić and V. Bobanac, "An accurate charging model of battery energy storage," *IEEE Transactions on Power Systems*, vol. 34, no. 2, pp. 1416–1426, 2019.

[8] K. N. Qureshi, A. Alhudhaif, and G. Jeon, "Electric-vehicle energy management and charging scheduling system in sustainable cities and society," *Sustainable Cities and Society*, vol. 71, p. 102990, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210670721002766

[9] Y. Lovcha, A. Perez-Laborda, and I. Sikora, "The determinants of co2 prices in the eu emission trading system," *Applied Energy*, vol. 305, p. 117903, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261921012162

[10] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou, "Information theoretic mpc for model-based reinforcement learning," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1714–1721.

[11] M. T. Lawder, B. Suthar, P. W. Northrop, S. De, C. M. Hoff, O. Leitermann, M. L. Crow, S. Santhanagopalan, and V. R. Subramanian, "Battery energy storage system (bess) and battery management system (bms) for grid-scale applications," *Proceedings of the IEEE*, vol. 102, no. 6, pp. 1014–1030, 2014.

[12] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*. PMLR, 2016, pp. 1928–1937.

[13] A. Wilcox, A. Balakrishna, J. Dedieu, W. Benslimane, D. Brown, and K. Goldberg, "Monte carlo augmented actor-critic for sparse reward deep reinforcement learning from suboptimal demonstrations," *Advances in Neural Information Processing Systems*, vol. 35, pp. 2254–2267, 2022.

[14] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*. PMLR, 2015, pp. 1889–1897.

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

[16] C. C.-Y. Hsu, C. Mendler-Dünner, and M. Hardt, "Revisiting design choices in proximal policy optimization," *arXiv preprint arXiv:2009.10897*, 2020.

[17] M. Pleines, M. Pallasch, F. Zimmer, and M. Preuss, "Generalization, mayhems and limits in recurrent proximal policy optimization," *arXiv preprint arXiv:2205.11104*, 2022.

[18] M. L. Bynum, G. A. Hackebeil, W. E. Hart, C. D. Laird, B. L. Nicholson, J. D. Siirola, J.-P. Watson, and D. L. Woodruff, *Pyomo–optimization modeling in python*, 3rd ed. Springer Science & Business Media, 2021, vol. 67.

[19] Stable-Baselines3, "Stable-Baselines3 Docs - Reliable Reinforcement Learning Implementations — Stable Baselines3 2.2.1 documentation," Online, Accessed 2023. [Online]. Available: https://stable-baselines3.readthedocs.io/en/master/

[20] A. Barreto, W. Dabney, R. Munos, J. J. Hunt, T. Schaul, H. P. van Hasselt, and D. Silver, "Successor features for transfer in reinforcement learning," *Advances in neural information processing systems*, vol. 30, 2017.