Towards Safe Model-Free Building Energy Management using Masked Reinforcement Learning

Sharath Ram Kumar, Rémy Rigo-Mariani, Benoit Delinchant, Arvind Easwaran

Towards Safe Model-Free Building Energy Management using Masked Reinforcement Learning

Sharath Ram Kumar*†, Rémy Rigo-Mariani*, Benoit Delinchant* and Arvind Easwaran†
*Univ. Grenoble Alpes, CNRS, Grenoble INP, G2Elab
Grenoble, France
†Nanyang Technological University
Singapore
Email: sharath.ramkumar@cnrsatcreate.sg

***Abstract*—Autonomous control of building energy resources including HVAC and battery storage systems has the potential to optimize operations and achieve objectives such as cost minimization. Existing approaches either require an explicit mathematical model of the building, or resort to simple rule-based controls (RBC) which may be sub-optimal. Model-free reinforcement learning (RL) is a promising method to overcome these limitations - however, it often requires a large number of interactions with the real environment before learning a functional policy. In this work, we investigate 'Action Masking', a technique to improve the learning efficiency of RL algorithms while respecting safety rules during the learning phase. Our solution achieves a cost reduction of 6% compared to a baseline rule-based controller, and also outperforms a popular transfer learning strategy. This suggests that model-free RL approaches are feasible and practical for problems in this domain.**

I. Introduction

As of 2021, building operations account for almost 30% of the global energy consumption, as well as for 27% of the emissions from the energy sector. A significant fraction of this consumption is already used to operate Heating, Ventilation and Air-Conditioning (HVAC) systems today, and the global demand in this sector is expected to surge over the next few years. On the other hand, the growing penetration of smart buildings and distributed energy resources into the energy grid represents an opportunity to efficiently manage this demand, using novel energy management systems (EMS). Control of energy resources based on these new technologies can be used to optimize operations, and achieve targets such as cost minimization, decarbonization, and demand response. [1]

Conventional methods for achieving this automation are either based on rule-based control (RBC), or use an explicit building model to set up a mathematical optimization problem. RBCs are widespread due to their simplicity - however, their performance is often limited since they are unable to adapt to changing environment dynamics. Model-based approaches, while offering good performance, are difficult to implement because of the need for a mathematical model of the building and its energy resources, as well as forecasts for exogenous variables such as weather and solar irradiance [2].

In this context, there is growing interest in model-free methods such as reinforcement learning (RL) for control tasks, as they offer the potential for high-performance, adaptive controllers without significant engineering effort. Deep Reinforcement Learning (DRL) is a sub-field of RL which has recently achieved a high level of success in complex tasks in fields such as computer science and robotics. A major issue hindering the adoption of DRL techniques in real-world tasks, such as Building Energy Management (BEM), is the requirement for a large number of interactions with the environment during the learning phase, which is only possible in a virtual environment or a simulation. To address this issue, recent research has focused on safe exploration for DRL, where guarantees such as constraint satisfaction are incorporated into existing algorithms. However, many of the proposed techniques still require a mathematical model [1].

In this paper, we demonstrate the use of a simple Action Masking approach to enforce constraints on a DRL agent during its training and deployment, without the use of a model of the environment. The key contributions of our work are, in the context of a building energy management problem:

- The use of Action Masking to embed constraints based on domain knowledge.
- A comparison of model-free RL approaches focused on safe exploration.

II. Problem Statement

A. Outline and Objectives

The control agent manages the cooling HVAC and the battery storage systems for a small office building situated in Singapore. The building is equipped with a 30 kWh Battery Electrical Storage System (BESS), capable of charging and discharging at 7.5 kW. Additionally, it has an off-grid solar energy source rated at 14 kWp. There is a 3-level pricing scheme in place, based on the time-of-use, as shown in Table I. In this study, both the BESS and the solar source are used only for local consumption - as such, there is no feed-in tariff for surplus generation.

The objective of the controller is to achieve the minimum electricity cost while maintaining a mean zone temperature of 25°C during occupied hours, i.e, between 7 am and 7 pm on

TABLE I
ELECTRICITY PRICING SCHEME USED IN THIS WORK

Time	Price (S$/kWh)
Holidays/Off-Peak	0.15
7 AM - 9 AM	0.25
9 AM - 2 PM	0.5
2 PM - 10 PM	0.25

working days. The latter is not a hard constraint - however, when comparing different controllers which incur the same cost, the better-performing controller is assumed to be the one that follows the setpoint better.

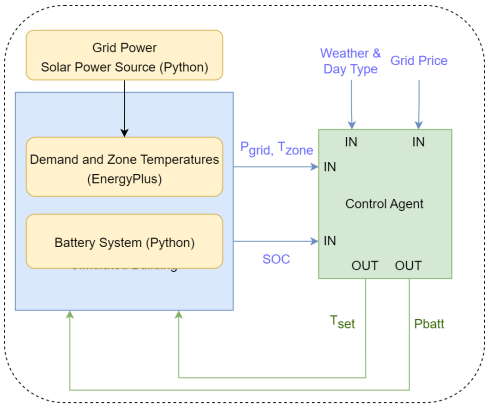

Fig. 1. Schematic Illustration of Control Problem

B. Control Actions

The controller is responsible for choosing the temperature setpoints for the cooling HVAC system, as well as the magnitude of charge and discharge of the battery system. The control timestep is 15 minutes. The key inputs are the grid power (P_{t-1}^{grid}) and the zone temperature (T_{t-1}^{zone}) at the end of the previous time step, alongside the battery state-of-charge (SOC) (B_{t-1}^{SOC}), the grid electricity price (p_t^g) and the type of day ($D_t^{holiday}$) for the current time step.

A schematic of the environment and the control actions are shown in Fig 1. The building is treated as a single zone building with a direct expansion cooling coil and condenser system. It is simulated using EnergyPlus (v22.2) and makes use of TMY weather data for Singapore. The other components (solar energy source and battery system) are simulated using Python, and communicates with the building simulation using the EnergyPlus Python API.

C. Baseline Controller

To evaluate the performance of different approaches, two baseline RBCs were developed based on a simple time-of-use strategy, which aim to charge the battery during off-peak hours. The actions of these controllers are summarized in Table II.

III. METHODS AND IMPLEMENTATION

The control problem is framed as a finite horizon Markov Decision Process (MDP), which is characterized by a state

TABLE II
RULE-BASED CONTROLLERS

Time	RBC 1		RBC 2	
	T_{set}(°C)	P_{bess}(kW)	T_{set}(°C)	P_{bess}(kW)
Before 6 AM	Off	-2.6	Off	-2.4
6 AM - 7 AM	25.85	-3	Off	-2.4
7 AM - 9 AM	25	1.5	25	0.75
9 AM - 11 AM	24.85	2.7	25	0.75
11 AM - 1 PM	25.15	2.7	25	0.75
1 PM - 2 PM	24.85	2.7	25	0.75
2 PM - 5 PM	25	0.55	25	0.75
5 PM - 7 PM	25.65	0.55	25	0.75
After 7 PM	Off	-2.6	Off	-2.4

transition function that depends only on the current state and the action, and a reward function which depends on the previous state, the action taken, and the next state [3]. A schematic representation of an MDP is shown in Fig 2. The task of an agent in an MDP is to take actions to maximize the net reward accumulated over the horizon.

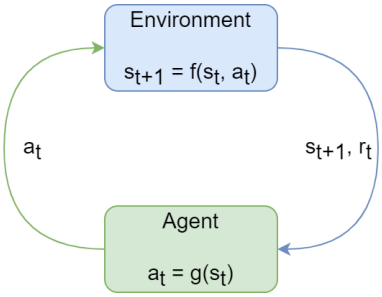

Fig. 2. Markov Decision Process

Recently, DRL has achieved notable success in solving MDPs with unknown transition dynamics, as is the case in the problem under consideration. In this paradigm, agents are implemented as neural networks, whose weights are updated by gradient ascent using a variant of the Bellman equation to maximize the expected future reward [4].

A. Proximal Policy Optimization (PPO)

PPO is an on-policy DRL algorithm which learns a stochastic policy by constraining the maximum deviation between subsequent policies during the weight update step [5]. Compared to other algorithms such as Deep-Q Networks (DQN), PPO is known to be more stable during the learning phase, and less dependent on hyperparameter tuning.

B. Action Masking

In reinforcement learning problems with large action spaces, a strategy employed to improve the exploration efficiency of the agent is to eliminate actions which are invalid or undesirable given a particular state [6]. In stochastic policies such as PPO, this is implemented by setting the probabilities of these actions to zero, and the process is called "Action Masking" [7].

A schematic of the working of an action mask is shown in Fig 3 - here, P_1 to P_3 represent the relative confidence the controller has in taking the corresponding action, and the mask

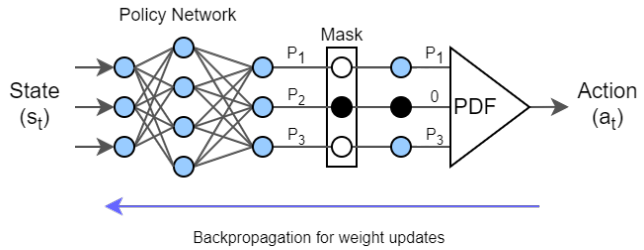

Fig. 3. Action Masking in Stochastic RL

eliminates the chance for a specific action (P_2) to be selected based on pre-existing rules. The action mask implemented in this work is based on prior knowledge of the problem, and enforces the following constraints:

1) If the battery is full (empty), eliminate actions that would charge (discharge) it.
2) The cooling system must be switched off after 10 PM and before 5 AM.
3) If the zone temperature is above 26.5°C during occupied hours, the cooling system cannot be switched off.

C. Behavioural Cloning

Behavioural cloning (BC) in reinforcement learning is a method to learn a policy by imitating the actions of an expert operating in the same environment. The most common approach is to implement a function approximator, such as a neural network, which is trained on a dataset of recorded expert interactions.

It should be noted that the expert in this scenario is taken to be the RBC, and the dataset is created by recording one year of interactions with the environment. First, a clone of the RBC is created by training a neural network to predict its actions given the states. This network is subsequently deployed in the real environment for online learning using a reward function and the PPO algorithm. A lower value for the PPO clip rate (0.2) was used in this step, as it represents a numerical measure of the maximum deviation between subsequent policies in the learning process - in other words, it discourages the agent from moving too far from the RBC policy in order to improve stability.

In a previous work, we showed that BC followed by online learning using PPO is an effective strategy for energy management problems, especially when the existing rule-based controller performs well [8]. The same technique is used in the present work to understand its relative strengths and weaknesses compared to reinforcement learning with an action mask.

D. DRL Problem Formulation

1) Train and Test Duration: The deployment period is set for a period of 3 years, from January 2015 to December 2017. There is an additional 30-day test set, which is used to evaluate the final strategy learnt by each method. This test set is made up of 15 days each in February and August 2020. Finally, a 1-year window between January 2014 and December 2014 is used to record the RBC operations for BC-based experiments. As the experiment is conducted using TMY weather data [9] from 2007-2021, the different divisions are implemented by setting the run period parameter in EnergyPlus.

2) State and Action Space: The state and action spaces available for the agent are summarized in the Table III. Here, t is the time step at which the controller takes an action. The state space consists of 6 variables, containing information about the BESS, the previous zone temperature, the electricity price and the grid consumption from the previous time step. Notably, the controller does not use any forecasts, or explicit information about the solar power generation, which represents an advantage compared to model-based approaches.

The action space is a 36-dimension discrete space offering 4 levels for the HVAC control, and 9 levels for the battery power control. The battery action linearly divides the interval [-7.5 kW, +7.5 kW] into 9 discrete points, with each one represented by one control action (a_{bess}). The 4 settings for the setpoint control (a_{hvac}) are {HVAC Off, 24°C, 25°C, 26°C}.

TABLE III
STATE AND ACTION SPACES

State Space	
Parameter	Timestep
Time of Day	t
Is Holiday?	t
Grid Price	t
Battery SOC	$t-1$
Zone Temperature	$t-1$
Grid Power	$t-1$

Discrete Action Space
$a_{bess} \in \{0, 1, .., 8\}$
$a_{hvac} \in \{0, 1, 2, 3\}$

The choice of constraints described in the section III-B was observed empirically to reduce the number of actions available to the agent by 47% on average.

3) Reward Function: The reward function used for the agent is the product of two quadratic terms, one representing the thermal comfort performance and the other representing the grid price performance. They are calculated using equations (1) - (3). The coefficients for the two terms are chosen to adequately penalize the agent for high costs or poor thermal comfort, while ensuring a maximum reward of 1.0.

$$r_t^{therm,A1} := -0.11\Delta T_t^2 - 0.22\Delta T_t + 0.89 \quad (1)$$

$$r_t^{grid} := 1.0 - 6(c_t^{grid})^2 \quad (2)$$

$$R_t = r_t^{grid} \times r_t^{therm} \quad (3)$$

Here, R_t is the reward value, ΔT_t is the difference between the zone temperature and the reference temperature, $c_t^{grid} (= P_t^{grid} \times p_t^g)$ is the cost incurred by the agent and the subscripts represent the timestep. If both components are negative, R_t is inverted to preserve the intended shape of the function. This formulation of the reward function is labelled *A1* in the results section, and shown in Fig 4.

To test the impact of different reward function shapes on the thermal comfort performance, the following alternate formulations in Eqs (4)-(5) (named *A2* and *A3* respectively) were also tested for the second term.

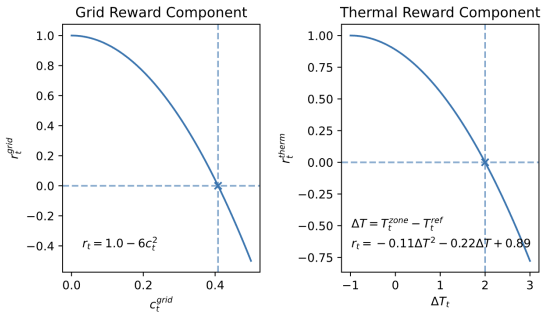

Fig. 4. Reward Function Components

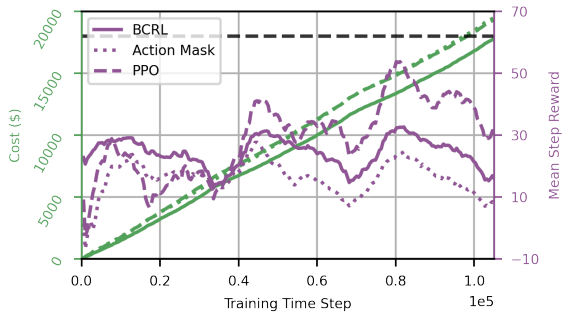

Fig. 5. Training Performance of Different Agents

$$r_t^{therm,A2} = 1 - max(\Delta T_t, 0)^2 \tag{4}$$

$$r_t^{therm,A3} = 1 - \Delta T_t^2 \tag{5}$$

Fig 5 shows the mean reward obtained (solid) by different agent types under reward $A1$, as well as the cumulative cost (dashed) during the training period. The black dashed line is the net cost incurred by the RBC in the same period.

4) Hyperparameters: All the Deep RL experiments performed in this work make use of the implementation of the PPO algorithm in the popular Python library, Stable-Baselines3 [10]. Table IV summarizes the key hyperparameters used - the specific values were determined using Optuna [11], an automated hyperparameter search program, through an search-and-prune algorithm over 50 iterations each for the continuous and the discrete action spaces. Additionally, due to the stoichastic nature of PPO, all reported values are averaged over 3 runs with different random seeds.

TABLE IV
HYPERPARAMETERS USED

Hyperparameter	Continuous	Discrete
Learning Rate	2.7e-4	2.3e-4
Gamma	0.999	0.988
GAE Lambda	0.99	0.840
Clip Range	0.6	0.7
Steps Per Update	12 days	5 days
Episode Length	1 day	1 day

IV. RESULTS AND DISCUSSION

The key metrics from each experiment are shown in Table V. The thermal comfort score is the probability that the indoor temperature is within 0.1°C of the setpoint during occupied hours. All values shown are calculated over the 30-day test set.

TABLE V
SUMMARY OF RESULTS

Agent	**Reward Function**	**Cost ($)**	**Comfort Score**
RBC 1	None	468.43	0.43
RBC 2	None	513.73	0.65
BC 1	None	482.15	0.52
BC 2	None	513.53	0.65
Direct RL	A1	428.89	0.09
Direct RL	A2	463.83	0.93
Direct RL	A3	476.70	0.93
Masked RL	A1	**439.27**	0.39
Masked RL	A2	563.76	0.44
Masked RL	A3	472.47	0.53
BC 1 + RL	A1	460.60	0.55
BC 1 + RL	A3	473.03	0.56
BC 2 + RL	A1	496.77	0.72

A. Overview and Pareto Front

The Pareto Front in Fig 6, plotted using the results in Table V, can be used to study the tradeoffs associated with each type of controller, and to understand the relative strengths and weaknesses. Fig 6 also contains a plot of the mean indoor temperature during the test period, as well as the grid power profile for different agents. The final strategy used by each approach can be visualized in Fig 7.

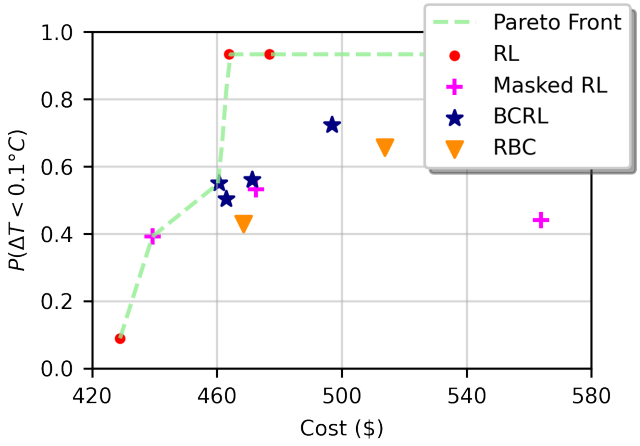

Fig. 6. Pareto Front Plot

The agents based on direct RL are able to outperform the reference RBC, but have a large variation in performance across different runs. While it may be possible to improve the stability of these agents by changing the hyperparameters of the training process, or modifying the reward function, such an approach is not feasible in practice for a real building energy management

problem. The indoor thermal comfort may not be respected at all during the training process as the agent is free to take any available action. This unpredictability during the training period is one of the major reasons hindering the deployment of DRL in critical control tasks.

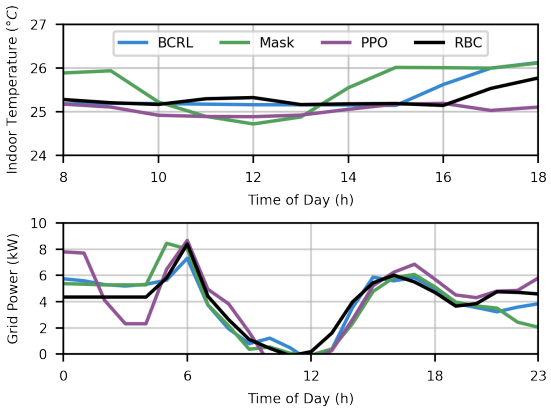

Fig. 7. Top: Average Indoor Temperature, Bottom: Average Grid Power Profile

B. Action Masking

The best performing agent for this task was obtained through action masking. Due to the constraints embedded in the mask, the agent is prevented from exploring parts of the action space that would result in poor thermal comfort while training. This is an important guarantee required to use black-box deep RL models in real-world control problems. It is notable that the agent converged to the best policy in the study through this safe exploration process, achieving a 6% reduction in energy costs compared to the reference RBC. However, the use of a mask does not remove the dependence on the reward function, as evidenced by the action masking agents trained using the other reward formulations, which incur a higher cost while maintaining thermal comfort.

C. Behavioural Cloning and Online Learning

BC + RL offers improved stability at the cost of significantly limiting the agent's exploration. As a result, the agents' performance is similar to the RBC, with only small improvements obtained through online learning. This is clearly visible in Fig 6, where the BC + RL runs are clustered close to their respective RBCs. This indicates that using online learning after BC is an effective strategy only if the underlying RBC performs well; in other words, the agent is unable to deviate sufficiently from the actions of the RBC without compromising the training stability. Compared to the other methods, BC + RL is also less dependent on the reward function used.

D. Impact of Problem Complexity

It is important to note that even for the single-zone building considered in the study, it was not feasible to directly use DRL algorithms to learn a control policy. In real applications, the environment is often more complex, with multiple interdependent zones and decision variables. Action masking is a reasonable strategy in this scenario to implement a DRL agent that offers safe exploration. It has been used successfully in more complex problems (such as games and autonomous driving) [6], given that the required constraints may be expressed using explicit rules.

V. Conclusion

Action Masking is a practical way to implement DRL algorithms such that the agents respect given rules during the training phase. This is a key requirement for the use of such methods in real-world applications such as energy management, where some safety guarantees are mandatory. While not fully eliminating the dependence on a well-defined reward function, the technique has the potential to outperform other model-free approaches, such as RBCs and transfer learning. Further research is required to assess the scalability of the solution to more complex problems in this domain.

VI. Acknowledgements

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme.

References

[1] K. Mason and S. Grijalva, "A review of reinforcement learning for autonomous building energy management," 2019.

[2] J. Drgoňa, J. Arroyo, I. C. Figueroa, D. Blum, K. Arendt, D. Kim, E. P. Ollé, J. Oravec, M. Wetter, D. L. Vrabie, and L. Helsen, "All you need to know about model predictive control for buildings," *Annual Reviews in Control*, vol. 50, pp. 190–232, 2020.

[3] M. L. Puterman, "Chapter 8 markov decision processes," in *Stochastic Models*, ser. Handbooks in Operations Research and Management Science. Elsevier, 1990, vol. 2, pp. 331–434.

[4] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, Nov. 2017.

[5] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 07 2017.

[6] S. Huang and S. Ontañón, "A closer look at invalid action masking in policy gradient algorithms," 2020.

[7] C.-Y. Tang, C.-H. Liu, W.-K. Chen, and S. D. You, "Implementing action mask in proximal policy optimization (ppo) algorithm," *ICT Express*, vol. 6, no. 3, pp. 200–203, 2020.

[8] S. R. Kumar, A. Easwaran, B. Delinchant, and R. Rigo-Mariani, "Behavioural cloning based rl agents for district energy management," in *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, ser. BuildSys '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 466–470.

[9] L. Lawrie and C. Drury, "Development of global typical meteorological years (tmyx)," 2022. [Online]. Available: http://climate.onebuilding.org

[10] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021.

[11] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," 2019.