Behavioural Cloning based RL Agents for District Energy Management

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

Behavioural Cloning based RL Agents for District Energy Management

Sharath Ram Kumar*
Univ. Grenoble Alpes, CNRS, Grenoble INP, G2Elab
Grenoble, France
Nanyang Technological University
Singapore
sharath.ramkumar@cnrsatcreate.sg

Arvind Easwaran
Nanyang Technological University
Singapore
arvinde@ntu.edu.sg

Benoit Delinchant
Univ. Grenoble Alpes, CNRS, Grenoble INP, G2Elab
Grenoble, France
benoit.delinchant@g2elab.grenoble-inp.fr

Remy Rigo-Mariani
Univ. Grenoble Alpes, CNRS, Grenoble INP, G2Elab
Grenoble, France
remy.rigo-mariani@g2elab.grenoble-inp.fr

ABSTRACT

In this work, we discuss a method to incorporate domain knowledge into a Reinforcement Learning (RL) environment through the process of behavioral cloning, in the context of a district energy management system. Prior knowledge, encoded into heuristic rule-based programs, is used to initialize a policy network for an RL agent, after which an RL algorithm is used to improve on this by optimizing against a reward function. The key ideas are implemented in the CityLearn framework, where the resulting controller is used to manage the electrical energy storage for 5 buildings in a district. We demonstrate that the resulting agents offer measurable performance gains compared to existing baselines, offering a 3.8% improvement over the underlying rule-based controller, and a 20% improvement over a pure RL based controller. We also illustrate the possibility of using imitation learning to develop agents with desirable characteristics without explicit reward shaping.

CCS CONCEPTS

• **Computing methodologies** → *Intelligent agents.*

KEYWORDS

Reinforcement Learning, Imitation Learning, District Energy Management

ACM Reference Format:
Sharath Ram Kumar*, Arvind Easwaran, Benoit Delinchant, and Remy Rigo-Mariani. 2022. Behavioural Cloning based RL Agents for District Energy Management. In *Third ACM SIGEnergy Workshop on Reinforcement Learning for Energy Management in Buildings & Cities (RLEM) (RLEM '22), November 9–10, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3563357.3566165

RLEM '22, November 9–10, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9890-9/22/11…$15.00
https://doi.org/10.1145/3563357.3566165

1 INTRODUCTION

Demand Response at the building level is widely considered as a promising avenue to improve the cost and energy efficiency of power grids. The field has grown in popularity in recent years due the growing penetration of distributed energy resources in the energy mix, such as solar panels and battery systems. However, the development of control systems for distributed building energy management remains a challenge due to the complex nature of buildings, which necessitates the use of detailed models and requires significant engineering effort [5]. Reinforcement Learning (RL) techniques are an active area of research in this context, since they can be used to build model-free controllers which learn by directly interacting with the target environment. Specifically, recent developments in the field of Deep RL, which combine traditional RL algorithms with deep neural networks, have proven effective in various continuous control tasks across diverse fields. RL solutions are typically not used in critical tasks in practice today due to the inherent unpredictability and the curse of dimensionality [6].

The CityLearn environment is a computer simulation framework based on the popular OpenAI Gym paradigm, and was developed to drive research in the field of Deep RL based Building Energy Management solutions. It uses a set of real-world datasets covering electricity demand, weather, and electricity pricing data, along with detailed models of various DERs such as electrical storage systems. Many RL-based controllers have been developed using CityLearn, based on different strategies related to the architecture and level of inter-agent communication. Conventionally, the operational effectiveness is reported relative to a reference Rule-Based Controller (RBC) - however, recent work has shown the drawbacks of this approach, demonstrating that a slightly modified RBC is able to achieve higher performance than most known RL-based solutions in CityLearn [7].

Imitation learning is an active area of research in the context of deep RL, as it holds the promise for agents to exploit existing knowledge and greatly speed up the training, while mitigating the need for complex reward function design. Behavioural Cloning is one such approach, where an agent is trained (typically through supervised learning) on data collected from the actions of an expert in the same environment. The technique has been used to

obtain promising results across a range of tasks related to robotic manipulation [8].

In this work, we propose a solution in which an RL-based controller can be developed utilizing a reference RBC. The approach uses the Proximal Policy optimization (PPO) learning algorithm [11] in combination with behavioural cloning. The use of imitation learning results in controllers with much more predictable actions and speeds up the training process, and a subsequent training step results in a small performance improvement over the underlying RBC. Notably, the learning process does not mandate the use a complex reward function, instead using a simple metric which is empirically known to be effective in the Citylearn environment.

2 RELATED WORK

The CityLearn environment has been used to study the use of reinforcement learning in the context of district energy management, and the potential of such techniques to develop purely data-driven, model-free controllers has been demonstrated [13]. Several controller architectures have been developed to solve this environment, with the performance of the resulting agents evaluated against a benchmark rule-based controller. Vazquez-Cantelli et al proposed MARLISA, a distributed framework in which each agent learns to predict its own energy consumption while taking control actions, with the aim of improving agent performance through information sharing [12]. Centralized controllers have also proven effective in solving the environment, where a single agent oversees the states and decides the actions for all buildings. A centralized agent based on the Soft Actor-Critic algorithm was developed by Kathirgamanathan, and was shown to exhibit good performance compared to the reference rule-based controller [3]. However, Nweye et al recently reported on the limitations of such approaches, and showed that an improved rule-based controller exhibited state-of-the-art performance in this environment [7].

Imitation learning, when combined with deep learning, has shown promise in a diverse set of problems. Hua recently reviewed the use of imitation and transfer learning in the field of robotics, especially for tasks such as manipulation and fine control [2]. Recently, a method to use imitation learning in the context of a building energy management problem was described by Gao et al [1], where the final agent learns to imitate the actions of a Model Predictive Controller (MPC) with a perfect forecast of the environment variables. Pezzotti et al developed MimicBot, a deep RL agent in a Fantasy Football game environment, which achieved excellent performance when RL training techniques were applied after an initial behaviour cloning step [9]. In a more advanced scenario, Liu et al created a framework to train simulated humanoid figures utilizing motion-capture data, and showed that complex behaviours requiring multi-agent coordination could be captured using this approach [4].

While previous work has focused on the application of deep RL and imitation learning in the context of building energy management systems (BEMS), we are not aware of any existing work where these techniques are applied in the context of a district energy management (DEM) problem. Traditional control methods used in BEMS, such as MPCs, are often not viable here due to the increased complexity caused by the presence of multiple buildings, resulting in the need to resort to heuristic approaches [10]. As such, the combination of IL and DRL presented in this work is of significance as a practical method to transition from these heuristic rules to a data-driven controller, without losing the knowledge encoded in the former approach.

3 IMPLEMENTATION

3.1 CityLearn 2022 Environment

The CityLearn 2022 Environment is an OpenAI gym environment which allows for the development and simulation of RL-based controllers for distributed energy management. The environment offers the possibility of choosing from up to 28 different observations for each building, covering both global parameters such as weather and building-specific data such as hourly demand and solar generation. Each building is equipped with a electric energy storage system, which can be charged or discharged at each time step by the controller. The performance of this controller is finally evaluated against that of a naive agent, which does not use the battery, on two metrics - net electricity price and net carbon consumption. In this study, we use only the net electricity price as a comparison metric. This choice is justified since the focus of the study is on discussing the use of behaviour cloning, and not on achieving the best scores in the simulation.

In the present work, data for a period of 1 year for 5 buildings is used - the models are trained on the data for the first 10 months, and evaluated on their performance for the subsequent 2 months. A centralized architecture is chosen for the controller, such that a single agent is able to observe the states for all 5 buildings, and take actions to charge or discharge the electric storage of each building at any time step. The observation space used in our formulation, made up of 22 data points for the whole district, is outlined in Table 1.

Table 1: State space for RL Problem Formulation

Parameter	Observation Type
Month	Shared
Hour	Shared
Direct Solar Irradiance	Shared
Direct Solar Irradiance (6h Forecast)	Shared
Carbon Intensity	Shared
Electricity Pricing	Shared
Electricity Pricing (6h Forecast)	Shared
Non-shiftable Load	Per Building
Solar Generation	Per Building
Electrical Storage SOC	Per Building

The above parameters were chosen from the set of all available information provided by the environment at every time step, omitting metrics related to the temperature, humidity and daylight savings status. This selection was done empirically, and adjusted based on the performance of the agents in the simulation. The chosen metrics are normalised based on their running mean and variance before being input to the policy network.

The reward function $r(t)$ at time step t is given below, where N is the number of buildings and e_i represents the net electricity cost for building i at time step t:

$$r(t) = -1 * (\sum_{i=1}^{N} \max(0, e_i))^2$$

The reward function used for the agent is purely based on the rectified net electricity price per time step, which is a simple metric based on the energy consumption and the electricity price. It should be emphasised that this choice was made to demonstrate the effectiveness of using behaviour cloning in encouraging positive behaviours for a given agent without explicitly rewarding them.

3.2 Algorithms

The learning process consists of two steps. First, the policy, represented by the weights of a neural network, is initialized by cloning the behaviour of a reference rule-based controller. This process is outlined in Algorithm 1. Next, the PPO algorithm is used to train this network further to generate the final policy, which typically achieves marginal gains over the RBC. The hyperparameters used in this work are listed in Table 2. It should be noted that a lower learning rate and clipping parameter were chosen for the second PPO training cycle, so that subsequent policies are not too different from the cloned policy. The RBC used in our work is outlined in Fig 1. The rules are structured such that the battery is charged during the late nights and early mornings, and discharged to meet the peak load.

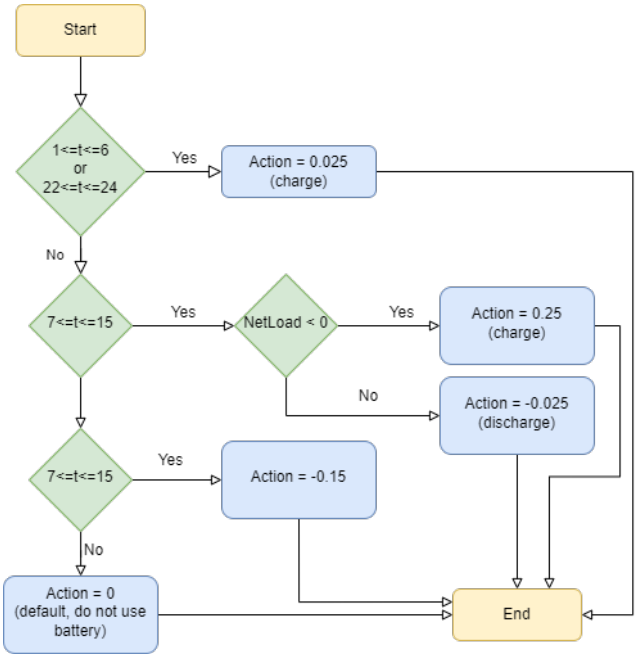

Figure 1: Rule-Based Agent Action Sequence

Algorithm 1 Behaviour Cloning

1: Initialize the rule-based policy π^R, target policy π_θ^{BC} and imitation buffer B
2: Initialize environment ε and initial state $s = s_0$
3: **while** B is not full **do**
4: Add $(s(t), \pi^R(s(t)))$ to B, and update $s(t+1) = \varepsilon(\pi^R(s(t))$
5: **end while**
6: Split B into B_{train}, B_{test}
7: **for** $i = 0, i < num\_training\_iters$ **do**
8: Sample mini-batch B_{mb} from B_{train}
9: Calculate Loss $\sum_{B_{mb}} \mathcal{L}(\pi^R(s(t)), \pi_\theta^{BC}(s(t)))$
10: Update $\theta := \theta - \gamma\nabla\mathcal{L}$
11: **end for**
12: Evaluate the model using B_{test}

The loss function $\mathcal{L}$ used in this implementation consists of a log-loss term which penalizes the difference between the model output and the ground truth, and a term to promote entropy in the stochastic policy. The first term addresses bias in the model output, while the latter term address variance.

Table 2: Hyperparameters Used In This Work

Parameter	PPO	BC + PPO	Description
lr_{BC}	None	$1e-4$	Learning Rate for Behavioural Cloning
e_{BC}	None	$1e-3$	Entropy Weight for Behavioural Cloning Loss
lr_{PPO}	$3e-4$	$1e-4$	Initial Learning Rate for the PPO Algorithm
$BatchSize$	256	256	Mini-batch size during PPO updates
$HiddenLayerDimensions$	64,64	64,64	Dimensions of the policy network hidden layers
c_{PPO}	0.2	0.1	PPO Clip Range
γ_{PPO}	0.99	0.99	PPO Future Rewards Coefficient
λ_{GAE}	0.95	0.95	PPO GAE Coefficient

4 RESULTS AND DISCUSSION

The trained agents are able to find a better policy than an agent which does not use electrical storage, and exhibit good performance

after training for about 2M steps on the training data. The scores achieved by each agent are listed in Table 3 - it must be noted that a lower score is desirable, since the figures represent cumulative costs. Fig 2 shows the training curves for the control PPO agent and the agent which started the PPO training process after behaviour cloning.

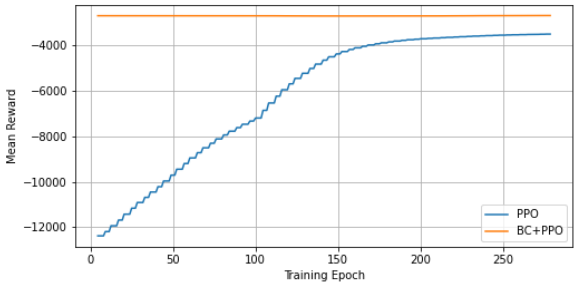

Figure 2: Training Rewards for the Agents

Table 3: Net Electricity Price for Different Agents

Agent Type	Whole Year ($)	Last 2 Months ($)
Baseline (No Storage)	8277.73	1614.90
Rule-Based Controller	6357.17	1281.82
PPO Agent (no BC)	7681.95	1535.99
Only BC (no PPO)	6136.06	1219.43
BC + PPO	**6113.04**	**1215.50**

It is visible from the training curves that the agent which starts out with a behaviour cloning prior does not exhibit a significant improvement in the training reward, especially when compared to the PPO-only agent. It is notable that such an approach still leads to small increase in the score for the final agent.

An important observation is that the actions of the agent trained using behaviour cloning are more approachable than those of the agent trained directly using PPO; for instance, the latter sometimes takes sub-optimal actions such as neglecting the energy storage system altogether for some buildings. The net energy demand for a period of 1 week, taken from the evaluation set, is shown in Fig 3.

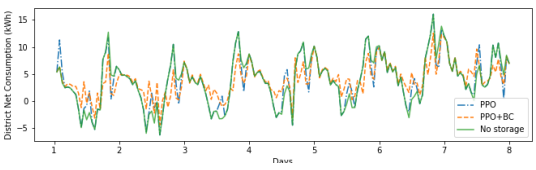

Figure 3: Net Electricity Consumption of the District

Due to the nature of the training process, the resulting agents take actions which closely resemble those of the underlying RBC. As the latter was developed manually using prior knowledge of the environment dynamics, the behavioural cloning step acts as a method to transfer existing knowledge into a neural-network based agent without constructing a complex reward function.

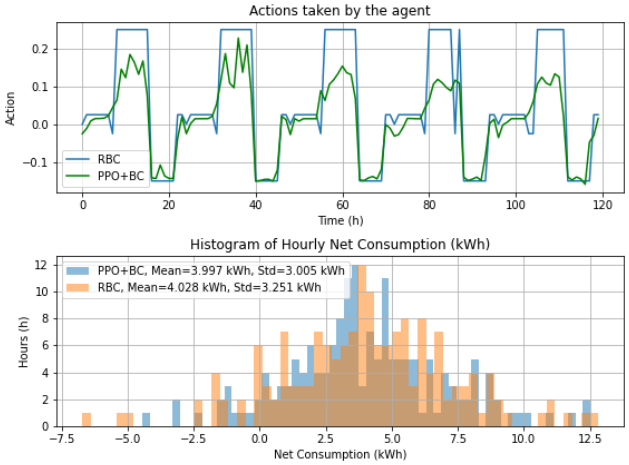

Figure 4: Comparison between RBC Agent and Final Agent

Fig 4 (top) shows a comparison between the actions taken by the RBC and the final agent under identical circumstances over a period of 1 week in the evaluation dataset for one of the buildings. It can be observed that the PPO+BC approach results in an agent that broadly follows the trends encoded in underlying rule based actor. For example, both controllers attempt to charge up the battery during off-peak hours (late night and early morning), and discharge during the day depending on the net load. The actions of the PPO+BC agent are slightly different in terms of magnitude of charging or discharging; these variations are introduced as a result of the data-driven training process. It can be seen from Fig 4 (bottom) that the net consumption is reduced as a result, which in turn reduces the net electricity price that is penalized in the PPO reward function. We can thus infer from Fig 4 that the improved performance quantified in Table 3 occurs as a result of the trained agent learning to make marginal changes to the actions of the RBC, in an attempt to maximize its reward.

Once the cloning step is completed, the resulting set of weights can be trained on-line in a continuous training operation. For instance, this opens up the possibility to develop functional controllers that constantly improve and adapt as new data is made available, which is highly desirable in this context. Since weight updation is a purely data-driven step, there is minimal additional effort required to implement the same.

An important caveat with the method presented here is that the performance of the final agent strongly depends on that of the initial rule-based agent. Training the clone agent using PPO only resulted in marginal gains compared to the underlying prior, since the hyperparameters chosen tend to restrict the exploration aspect of the learning process in exchange for stability. Further research is required to determine whether this limitation can be overcome by a different choice of hyperparameters.

5 CONCULSION AND FUTURE SCOPE

In this paper, we explore the possibility of embedding prior knowledge into a Deep RL based control system using imitation learning, in the context of the building energy management problem presented in the CityLearn environment. The resulting agents are able to translate a hard-coded set of rules into the weights of a neural network, and achieve marginal gains over the underlying policy through continuous training. The possibility of adapting the method to implement a controller that is training on-line using real-time data was discussed. It was noted that the technique allows for the training of viable agents without designing a complex reward function - however, the performance was observed to be strongly dependent on, and limited by, the performance of the initial heuristic agent. Future work may focus on achieving a significant performance improvement over the underlying controller, through hyperparameter tuning or by using a different reward function.

6 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] Shuhua Gao, Cheng Xiang, Ming Yu, Kuan Tak Tan, and Tong Heng Lee. 2022. Online Optimal Power Scheduling of a Microgrid via Imitation Learning. *IEEE Transactions on Smart Grid* 13, 2 (March 2022), 861–876. https://doi.org/10.1109/tsg.2021.3122570

[2] Jiang Hua, Liangcai Zeng, Gongfa Li, and Zhaojie Ju. 2021. Learning for a robot: Deep reinforcement learning, imitation learning, transfer learning. *Sensors* 21, 4 (2021), 1278.

[3] Anjukan Kathirgamanathan, Kacper Twardowski, Eleni Mangina, and Donal Finn. 2020. A Centralised Soft Actor Critic Deep Reinforcement Learning Approach to District Demand Side Management through CityLearn. (2020). https://doi.org/10.48550/ARXIV.2009.10562

[4] Siqi Liu, Guy Lever, Zhe Wang, Josh Merel, S. M. Ali Eslami, Daniel Hennes, Wojciech M. Czarnecki, Yuval Tassa, Shayegan Omidshafiei, Abbas Abdolmaleki, Noah Y. Siegel, Leonard Hasenclever, Luke Marris, Saran Tunyasuvunakool, H. Francis Song, Markus Wulfmeier, Paul Muller, Tuomas Haarnoja, Brendan Tracey, Karl Tuyls, Thore Graepel, and Nicolas Heess. 2022. From motor control to team play in simulated humanoid football. *Science Robotics* 7, 69 (2022), eabo0235. https://doi.org/10.1126/scirobotics.abo0235 arXiv:https://www.science.org/doi/pdf/10.1126/scirobotics.abo0235

[5] Jose Medina, Nelson Muller, and Ilya Roytelman. 2010. Demand response and distribution grid operations: Opportunities and challenges. *IEEE Transactions on Smart Grid* 1, 2 (2010), 193–198.

[6] Thanh Thi Nguyen, Ngoc Duy Nguyen, and Saeid Nahavandi. 2020. Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications. *IEEE transactions on cybernetics* 50, 9 (2020), 3826–3839.

[7] Kingsley Nweye, Bo Liu, Peter Stone, and Zoltan Nagy. 2021. Real-world challenges for multi-agent reinforcement learning in grid-interactive buildings. https://doi.org/10.48550/ARXIV.2112.06127

[8] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, Jan Peters, et al. 2018. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics* 7, 1-2 (2018), 1–179.

[9] Nicola Pezzotti. 2021. MimicBot: Combining Imitation and Reinforcement Learning to win in Bot Bowl. https://doi.org/10.48550/ARXIV.2108.09478

[10] Mohammad Sameti and Fariborz Haghighat. 2017. Optimization approaches in district heating and cooling thermal network. *Energy and Buildings* 140 (April 2017), 121–130. https://doi.org/10.1016/j.enbuild.2017.01.062

[11] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[12] Jose R. Vazquez-Canteli, Gregor Henze, and Zoltan Nagy. 2020. MARLISA: Multi-Agent Reinforcement Learning with Iterative Sequential Action Selection for Load Shaping of Grid-Interactive Connected Buildings. In *Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation* (Virtual Event, Japan) *(BuildSys '20)*. Association for Computing Machinery, New York, NY, USA, 170–179. https://doi.org/10.1145/3408308.3427604

[13] José R. Vázquez-Canteli and Zoltán Nagy. 2019. Reinforcement learning for demand response: A review of algorithms and modeling techniques. *Applied Energy* 235 (Feb. 2019), 1072–1089. https://doi.org/10.1016/j.apenergy.2018.11.002