Deep Reinforcement Learning for Energy Management in Residential Housing

Tim Walter [1]

Abstract

This paper provides a novel formulation of the carbon emission minimization problem within a smart home. The environment includes an energy storage system, a heat pump, and a flexible demand. The environment is formulated as a Markov Decision Process and tested using state of the art reinforcement learning methods. The effectiveness of the algorithms is compared to rule-based controllers. In addition, three variations of the environment are explored in an effort to improve the formulation in terms of learnability.

1. Introduction

The operations of buildings account for 30% of global final energy consumption and 26 % of global energy-related greenhouse gas emissions (IEA, 06.01.2024). This makes the optimization of energy consumption a major concern for environmental sustainability. This paper investigates the application of Deep Reinforcement Learning (DRL) to streamline energy usage within a single-family home, leveraging solar panels, electric batteries, heat pumps, and flexible demand mechanisms. The primary aim is to reduce the carbon footprint associated with housing operations.

2. Related Work

The application field of smart energy management encompasses a wide range of problems, such as heating and cooling (Blum et al., 2021)(Thomas Schreiber et al., 2020), flexible demand response (Jin et al., 2021) and energy storage (Nakabi & Toivanen, 2021). Moreover, the scale and level of detail of the tackled problems vary greatly, going from single buildings (Blum et al., 2021), to microgrids (Nakabi & Toivanen, 2021) (Castellanos et al.), to eventually continent-wide electricity grids (Hörsch et al., 2018). The two most promising approaches to solve such problems are classical Model Predictive Control (MPC) (Basantes et al., 2023) and Reinforcement Learning (RL) (Nakabi & Toivanen, 2021) (Jin et al., 2021) (Thomas Schreiber et al., 2020)(Zhu et al., 2022). Furthermore, there are also hybrid approaches that combine the two methods (Javier Arroyo et al., 2022).

This paper is mainly inspired by (Nakabi & Toivanen, 2021), who describe a microgrid environment with generation, storage, and consumers, where they optimize electricity pricing using various DRL algorithms.

3. Environment

The main contribution of this work is an environment modeling a single-family household, consisting of a Rooftop Solar Array (RSA), an Energy Storage System (ESS), a Thermostatically Controlled Load (TCL), and a Flexible Demand Response (FDR). The environment is a mixture of replayed data and dynamic components. It is built on the Gymnasium framework (Towers et al., 2023) and available on GitHub[1]. The main objective is minimizing the CO2eq emissions. The task is split into week-long episodes with hourly resolution, to capture daily and weekly patterns of demand and generation.

The RL paradigm requires the formulation as a Markov Decision Process, consisting of a transition function, action and observation spaces, and a reward function. The transition function is given by the replayed data and the dynamics of the components in the following. The observation space is a bounded subset of $\mathbb{R}^{11+H}$, with the observables given in Table 1. The action space is summarized in Table 2. The reward function is described in Section 3.5.

3.1. Replayed Data

The data was either initially given in hourly resolution or down-sampled by averaging. The most relevant observables are displayed exemplary in Figure 1.

The auxiliary information included weather and time

[1]Department of Scientific Computing, Technical University of Munich, Munich, Germany. Correspondence to: Tim Walter <tim.walter@tum.de>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

[1]https://github.com/TimWalter/smart-energy-controller

data, which provided means for the agent to predict the carbon intensity of the electricity mix and the heating demand. It was sourced from the Photovoltaic Geographical Information System (PVGIS) (Thomas Huld et al., 2012).

The RSA was simulated using the PVLIB library (F. Holmgren et al., 2018), with panel specifications from the CEC database (Dobos, 2012)(Boyson et al., 2007).

The household's energy demand was sourced from the UCI archive (Georges Hebrail). Since the consumption was unraveled for different rooms, the kitchen, electric water heater and air conditioner were modeled as inflexible demand, while the laundry room as flexible demand.

The direct carbon intensity of the electricity mix at any given time was sourced from Electricity Maps (ElectricityMaps, 29.12.2023).

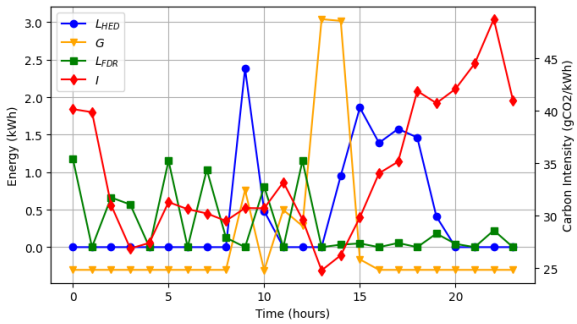

Figure 1. Replayed data for the first 24 hours.

3.2. Energy Storage System

The ESS is connected to the grid and the RSA. The charge is subject to the following dynamics

$$B_t = \max\{0, B_{t-1} - D_s + C_t\sqrt{\nu} - \frac{D_t}{\sqrt{\nu}}\}. \quad (1)$$

Here, $B_t \in [0, B_{max}]$ is the charge at time t and B_{max} is the capacity. Furthermore, ν denotes the round trip efficiency, D_s the self-discharge rate, $C_t \in [0, C_{max}]$ the charge rate with maximum C_{max}, and $D_t \in [0, D_{max}]$ the discharge rate with its respective maximum D_{max}.

3.3. Flexible Demand Response

The FDR can be influenced stochastically in a running time window of length H by a signal $a_{fdr,t} \in [-1, 1]^H$. Whether the power is consumed is determined by a Bernoulli process, with probabilities

$$p_t = \text{clip}\left(s + a_{fdr,t} * \exp\left(\frac{-1}{\beta}|t - t_s|\right), 0, 1\right), \quad (2)$$

where $s \in \{0, 1\}^H$ indicates the desired consumption with elements $s_i = \begin{cases} 1 & \text{if } t_i \leq t \\ 0 & \text{else} \end{cases}$, β is a patience parameter, and $t_s \in \mathbb{R}^H$ is the desired consumption time. If consumption is delayed more than once, $\frac{2}{H-1}$ of the power is consumed in each time step to ensure that all power has been consumed after the time window. This aims to facilitate learning by preventing infinite delays or a large correction at episode termination.

3.4. Thermostatically Controlled Load

The TCL encapsulates all devices that aim to maintain the temperature of a given heat mass. To utilize such loads as energy storage, the temperature of the heat mass is allowed to fluctuate within a given range. The TCL is modeled as a second-order system (Sonderegger, 1978) with the following dynamics:

$$\begin{aligned} T_t = T_{t-1} &+ \frac{1}{r_a}(T_{a,t} - T_{t-1}) \\ &+ \frac{1}{r_b}(T_{b,t} - T_{t-1}) \\ &+ \frac{1}{r_h} L_{TCL} a_{tcl,t} + q, \end{aligned} \quad (3)$$

where T_t is the indoor temperature at time t, r_a is the thermal mass of the air, $T_{a,t}$ is the current outdoor temperature, r_b is the thermal mass of the building material, and $T_{b,t}$ is the current building mass temperature, that evolves with

$$T_{b,t} = T_{b,t-1} + \frac{1}{r_b}(T_{t-1} - T_{b,t-1}), \quad (4)$$

$\frac{1}{r_h}$ is the power-to-heat coefficient, L_{TCL} is the nominal power, $a_{tcl_t} \in [-1, 1]$ is the heating signal, and q is the unintended heat drift. The heating signal is constrained by the desired temperature range, enforced through a backup controller as follows:

$$a_{tcl,t} = \begin{cases} -1 & \text{if } T_t \geq T_{max} \\ 1 & \text{if } T_t \leq T_{min} \\ a_{tcl,t} & \text{else,} \end{cases} \quad (5)$$

where T_{max} and T_{min} define the desired temperature range.

3.5. Reward Function

CO2eq emissions are modeled naively as carbon intensity I_t times produced energy E_p minus consumed energy E_c. The given objective is augmented by a temperature discomfort penalty DC to bias towards the desired temperature. The

Table 1. Observation Space.

OBSERVABLE	SYMBOL	REPLAYED
CARBON INTENSITY	I	X
HOUSEHOLD ENERGY DEMAND	L_{HED}	X
ROOFTOP SOLAR GENERATION	G	X
ESS CHARGE	B	
FDR POWER IN WINDOW	L_{FDR}	
INDOOR TEMPERATURE	T	
TIME STEP	t	X
MONTH OF YEAR	MoY	X
SOLAR IRRADIATION	SI	X
SOLAR ELEVATION	SE	X
OUTDOOR TEMPERATURE	T_a	X
WIND SPEED	WS	X

Table 2. Action Space.

ACTION	MIN. MEANING	MAX. MEANING
a_{ess}	DISCHARGE	CHARGE
a_{fdr}	DELAY	EXPEDITE
a_{tcl}	COOLING	HEATING

resulting reward function is

$$r_t = I_t(E_p - E_c) - DC \tag{6}$$

$$E_c = L_t + \frac{1}{r_h} L_{TCL} a_{tcl,t} + C_t + \sum_{p \in U_e} p_r + \sum_{p \in U_d} \frac{2p}{H-1} \tag{7}$$

$$E_p = G_t + D_t \tag{8}$$

$$DC = \delta \exp(|T_t - \frac{T_{max} + T_{min}}{2}|), \tag{9}$$

where U_e is the set of consumed FDR, p_r is the remaining power after discounting, U_d is the set of delayed FDR, and δ the discomfort coefficient.

Given that the task formulation is episodic, it differs from the real-world problem which has an infinite horizon. In the real-world problem, the initial state of the upcoming week is influenced by the past week. Therefore, to accurately reflect this, a correction is required at termination to evaluate the remaining state. The terminal reward is corrected by

$$reward \mathrel{+}= I_t B_t \tag{10}$$

$$reward \mathrel{+}= I_t(-\sum_{p \in U} p_r) \tag{11}$$

$$reward \mathrel{+}= I_t(-r_h \left| T_t - \frac{T_{max} + T_{min}}{2} \right|) \tag{12}$$

where U is the set of remaining FDR.

4. Experiments

As non-learning baselines an idle policy, which would always return zero actions, whose cumulative reward in the first episode is shown in Figure 2, and a thresholding policy, see Appendix B, were implemented.

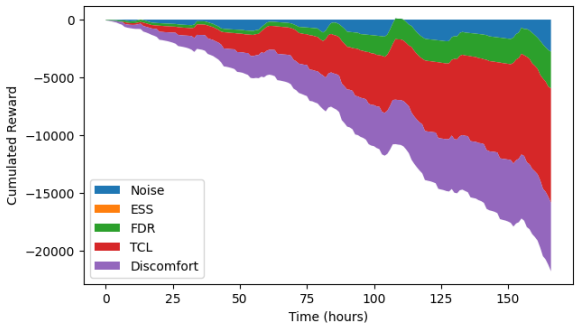

Figure 2. Cumulative reward for the idle policy.

As learning agents, Soft Actor Critic (SAC) (Haarnoja et al.) and Proximal Policy Optimization (PPO) (Schulman et al.) were chosen. Both implementations were taken from stable baselines 3 (Antonin Raffin et al., 2021), for more details see Appendix C and Appendix D.

Since RL algorithms are known to be sample inefficient (Irpan, 2018), the environment was initially restricted to a single episode in an effort to accelerate convergence. The training curve is shown in Figure 3 and all results in Table 3. Rewards were split by origin, as in Appendix F, to allow evaluation by task. The results were convergent but not satisfactory, since the agents did not outperform the thresholding baseline. The agents were unable to utilize the ESS effectively, while TCL and FDR actions were also far from optimal.

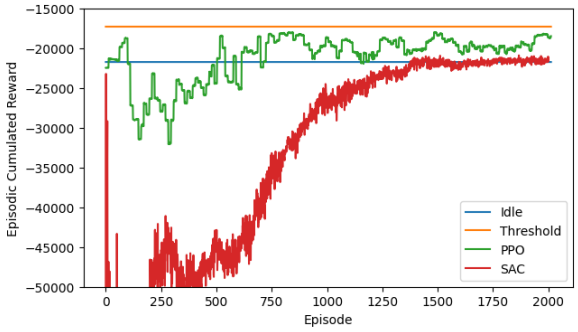

Figure 3. Training curve replaying a single episode.

To increase the learnability of the environment three variations were tested, further details can be found in Appendix E.

Table 3. Accumulated rewards split by origin per setting. Evaluation was done after every epoch and the best result is shown.

EXPERIMENT	ALGORITHM	ESS	FDR	TCL	DISCOMFORT	TOTAL
INITIAL SETTING	IDLE	0.00	-3173.56	-9887.61	-5912.63	-21746.74
	THRESHOLD	**633.24**	-3068.56	**-10988.57**	**-1106.55**	**-17303.38**
	PPO	0.00	**-3038.80**	-10749.44	-1423.62	-17984.79
	SAC	-82.24	-3063.11	-10036.19	-4958.21	-20912.68
STACKED CARBON INTENSITY	IDLE	0.00	-3173.56	-9887.61	-5912.63	-21746.74
	THRESHOLD	**633.24**	-3068.56	-10988.57	-1106.55	**-17303.38**
	PPO	0.00	**-3017.24**	**-10810.46**	**-1180.09**	-17780.73
	SAC	-18.01	-3077.25	-10226.50	-3403.34	-19498.02
EASED EXPLORATION	IDLE	0.00	-3173.56	-9887.61	-5912.63	-21746.74
	THRESHOLD	**633.24**	-3056.79	**-10988.57**	**-1106.55**	**-17290.67**
	PPO	0.00	-3034.17	-10055.01	-5072.91	-20933.18
	SAC	495.07	**-2952.68**	-12908.10	-4582.26	-22720.31
TERMINAL REWARD	IDLE	0.00	-3173.56	-9887.61	-5912.63	-21746.74
	THRESHOLD	**633.24**	**-3068.56**	**-10988.57**	**-1106.55**	**-17303.38**
	PPO	362.88	-3125.46	-10759.23	-1587.45	-17881.28
	SAC	-92.2	-3123.76	-13036.89	-3371.14	-22395.99

4.1. Stacked Carbon Intensity

To effectively act on any sub-task, the current carbon intensity has to be evaluated against the expected future. If the agents would fail to recognize low carbon intensity, their poor performance across sub-tasks would be explained.

To ease this task the carbon intensity observable was stacked into the future. While this improved the overall performance marginally, the overall issue remained.

4.2. Eased Exploration

The action space in the given setting is continuous and high dimensional, due to the dependence on H. In combination with the stochasticity of FDR control and the noisy reward discovering effective action sequences is challenging.

To shrink the action space, the FDR control was collapsed into a scalar. Moreover, actions were discretised for PPO and the agents were trained separately for each sub-task. The FDR control was also changed to deterministically control the consumption. This assumes that the optimal policy is invariant to U_e as long as the sum is maintained and that $\beta = \infty$. In addition, the RSA and household energy demand were removed from the observables and reward calculation.

The separate training enabled SAC to utilize the ESS effectively, while performance on the other sub-tasks remained poor. PPO performed significantly worse, which is probably due to the discretisation of the action space.

4.3. Terminal Reward

Another challenge of the setting is the reward structure for charging the ESS, expediting the FDR and heating or cooling the TCL. The rewards of those actions always occur delayed and only if the respective counter action is performed at a higher carbon intensity. Moreover, the rewards might even be deceptive, since those actions are necessary for an effective policy but always yield immediate negative rewards.

To remove the temporal reward-action tie, the reward was transferred to an accumulating observable and only actuated in the terminal state, which should encourage the agent to evaluate the entire trajectory of an episode holistically. This enabled PPO to effectively utilize the ESS, however the performance on all other sub-tasks worsened. Moreover, SAC performed significantly worse across sub-tasks.

5. Future Work

This paper provides an open challenge with a novel formulation of the problem of carbon intensity minimization in a smart home environment. To further enhance control, future work can explore model-based learning techniques, which could utilize the rich and aligned reward calculation. Moreover, imitation learning from rule-based controllers should guide the optimization and improve sample efficiency. Another promising approach to the problem is trying to predict the carbon intensity accurately for the desired period and then optimizing the control sequence for that period classically.

References

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL http://jmlr.org/papers/v22/20-1364.html.

Basantes, J. A., Paredes, D. E., Llanos, J. R., Ortiz, D. E., and Burgos, C. D. Energy management system (ems) based on model predictive control (mpc) for an isolated dc microgrid. *Energies*, 16(6):2912, 2023. doi: 10.3390/en16062912.

Blum, D., Arroyo, J., Huang, S., Drgoňa, J., Jorissen, F., Walnum, H. T., Chen, Y., Benne, K., Vrabie, D., Wetter, M., and Helsen, L. Building optimization testing framework (boptest) for simulation-based benchmarking of control strategies in buildings. *Journal of Building Performance Simulation*, 14(5):586–610, 2021. ISSN 1940-1493. doi: 10.1080/19401493.2021.1986574.

Boyson, W. E., Galbraith, G. M., King, D. L., and Gonzalez, S. Performance model for grid-connected photovoltaic inverters. 2007. doi: 10.2172/920449. URL https://www.osti.gov/biblio/920449,journal=.

Castellanos, J., Correa-Florez, C. A., Garcés, A., Ordóñez-Plata, G., Uribe, C. A., and Patino, D. An energy management system model with power quality constraints for unbalanced multi-microgrids interacting in a local energy market. URL http://arxiv.org/pdf/2212.01910.pdf.

Dobos, A. P. An improved coefficient calculator for the california energy commission 6 parameter photovoltaic module model. *Journal of Solar Energy Engineering*, 134 (2), 2012. ISSN 0199-6231. doi: 10.1115/1.4005759.

ElectricityMaps. Electricity maps — reduce carbon emissions with actionable electricity data, 29.12.2023. URL https://www.electricitymaps.com/.

F. Holmgren, W., W. Hansen, C., and A. Mikofski, M. pvlib python: a python package for modeling solar energy systems. *Journal of Open Source Software*, 3(29):884, 2018. doi: 10.21105/joss.00884.

Georges Hebrail, A. B. Individual household electric power consumption.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. URL http://arxiv.org/pdf/1801.01290.pdf.

Hörsch, J., Hofmann, F., Schlachtberger, D., and Brown, T. Pypsa-eur: An open optimisation model of the european transmission system. *Energy Strategy Reviews*, 22:207–215, 2018. ISSN 2211467X. doi: 10.1016/j.esr.2018.08.012. URL http://arxiv.org/pdf/1806.01613.pdf.

IEA. Buildings - energy system - iea, 06.01.2024. URL https://www.iea.org/energy-system/buildings.

Irpan, A. Deep reinforcement learning doesn't work yet, 2018.

Javier Arroyo, Carlo Manna, Fred Spiessens, and Lieve Helsen. Reinforced model predictive control (rl-mpc) for building energy management. *Applied Energy*, 309:118346, 2022. ISSN 0306-2619. doi: 10.1016/j.apenergy.2021.118346. URL https://www.sciencedirect.com/science/article/pii/S0306261921015932.

Jin, L., Chen, Z., Li, J., and Ye, T. Reinforcement learning method based load shifting strategy with demand response. In *2021 40th Chinese Control Conference (CCC)*, pp. 1586–1591, 2021. doi: 10.23919/CCC52363.2021.9549975.

Nakabi, T. A. and Toivanen, P. Deep reinforcement learning for energy management in a microgrid with flexible demand. *Sustainable Energy, Grids and Networks*, 25: 100413, 2021. ISSN 23524677. doi: 10.1016/j.segan.2020.100413.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. URL http://arxiv.org/pdf/1707.06347.pdf.

Sonderegger, R. C. *Dynamic models of house heating based on equivalent thermal parameters*. PhD thesis, Princeton University, New Jersey, 1978.

Thomas Huld, Richard Müller, and Attilio Gambardella. A new solar radiation database for estimating pv performance in europe and africa. *Solar Energy*, 86(6):1803–1815, 2012. ISSN 0038-092X. doi: 10.1016/j.solener.2012.03.006. URL https://www.sciencedirect.com/science/article/pii/S0038092X12001119.

Thomas Schreiber, Sören Eschweiler, Marc Baranski, and Dirk Müller. Application of two promising reinforcement learning algorithms for load shifting in a cooling supply system. *Energy and Buildings*, 229:110490, 2020. ISSN 0378-7788. doi: 10.1016/j.enbuild.2020.110490. URL https://www.sciencedirect.com/science/article/pii/S0378778820320922.

Towers, M., Terry, J. K., Kwiatkowski, A., Balis, J. U., de Cola, G., Deleu, T., Goulão, M., Kallinteris, A., KG, Arjun, Krimmel, M., Perez-Vicente, R., Pierré, A., Schulhoff, S., Tai, J. J., Shen, A. T. J., and Younis, O. G. Gymnasium, 2023. URL https://zenodo.org/record/8127025.

Zhu, D., Yang, B., Liu, Y., Wang, Z., Ma, K., and Guan, X. Energy management based on multi-agent deep reinforcement learning for a multi-energy industrial park. *Applied Energy*, 311:118636, 2022. ISSN 0306-2619. doi: 10.1016/j.apenergy.2022.118636. URL http://arxiv.org/pdf/2202.03771.pdf.

A. Further Environment Details

The replayed data was mainly recorded or simulated in the time from 07.01.2007 to 28.12.2008. Although some episodes had to be left out due to missing data, there were more than 100 episodes, of which 95 were supposed to be used for training, 4 for testing and the rest for evaluation. The power was standardized to kilowatts and the energy respectively to kWh. The location of the house was in France near Paris.

The carbon intensity data was only available from 2021 and 2022 for France, so the carbon intensity might be biased since the electricity mix might have drifted over the years. However, the daily, weekly and yearly frequencies should still be captured, although potentially drifted. The unit of carbon intensity is grams of CO_2 equivalent emissions per kWh.

To prevent the ESS from overcharging or overdischarging, the rates were further constrained by the current charge and capacity, such that actually $C_t \in [0, \min\{C_{max}, \frac{B_{max}-B_t+D_s}{\sqrt{\nu}}\}]$ and $D_t \in [0, \min\{D_{max}, (B_t - D_S)\sqrt{\nu}\}]$.

B. Thresholding Baseline

$$a_t = \begin{cases} \begin{bmatrix} 1 & [1]^H & \psi \end{bmatrix} & \text{if } I_t < \phi_1 \\ \begin{bmatrix} -1 & [0]^H & \psi \end{bmatrix} & \text{if } I_t > \phi_2 \\ \begin{bmatrix} 0 & [0]^H & \psi \end{bmatrix} & \text{else,} \end{cases} \tag{13}$$

where $\phi_1 = 65\frac{g}{kWh}$ and $\phi_2 = 85\frac{g}{kWh}$ are the lower and upper threshold respectively, while

$$\psi = sign(\frac{T_{max} + T_{min}}{2} - T_t)clip(\frac{T_t - \frac{T_{max}+T_{min}}{2}}{1.3}, 0, 1). \tag{14}$$

C. SAC

SAC is a squashed Gaussian policy, that employs entropy regularization, which adds a bonus reward in each time step proportional to the current entropy of the policy in an aim to encourage exploration. Furthermore, two Q-functions are learned and their minimal estimate is used to update the policy. The objective for SAC is

$$\max_\theta \mathop{\mathbb{E}}_{\substack{s\sim\mathcal{D} \\ \xi\sim\mathcal{N}}} \left[\min_{j=1,2} Q_{\phi_j}\left(s, \tilde{a}_\theta(s,\xi)\right) - \alpha H(\pi_\theta \mid s) \right], \tag{15}$$

where θ denotes the policy parameter, ξ normal Gaussian noise, $\tilde{a}_\theta(s,\xi)$ the action sampled from the current policy, α the entropy regularization coefficient, and $H(\pi_\theta \mid s) = \log \pi_\theta\left(\tilde{a}_\theta(s,\xi) \mid s\right)$ the entropy of the policy. The action samples are obtained as $\tilde{a}_\theta(s,\xi) = tanh(\mu_\theta(s) + \sigma_\theta(s) \odot \xi)$.

D. PPO

PPO is a trust-region based policy gradient method, which solves a constrained policy update policy via SGD. It achieves the trust region remarkably simple, by employing the following update objective:

$$\theta_{k+1} = \arg\max_\theta \mathop{\mathbb{E}}_{\substack{s\sim\mathcal{D} \\ a\sim\pi_{\theta_k}}} \left[L(s,a)\right] \tag{16}$$

$$L(s,a) = \min\left(\left| \frac{\pi_\theta(a \mid s)}{\pi_{\theta_k}(a \mid s)} \right|, 1 \pm \epsilon \right) A^{\pi_{\theta_k}}(s,a), \tag{17}$$

where $A^{\pi_{\theta_k}}(s,a)$ is the advantage of the current policy, and ϵ the trust region hyperparameter, clipping the update size. Alternatively, PPO can also be implemented using a KL-divergence constraint.

E. Training Procedure

All experiments were run for 2000 episodes with fixed initial conditions, this took around 4 hours per algorithm on one RTX 2060 super. The training parameters are shown in Table 4.

Table 4. Training Parameters

PARAMETER	VALUE
B_{max}	13.5
C_{max}	1
D_{max}	1
ν	0.95
D_s	0.01
B_0	0
H	25
β	20
δ	5
T_0	20
$T_{b,0}$	20
r_a	0.04
r_b	0.1
r_h	0.05
q	0.05
L_{TCL}	5
T_{max}	23
T_{min}	18

For the stacked carbon intensity, the next 3 hours of carbon intensity were given in the observable too.

For the eased exploration setting, the fdr observations were also only reported as a sum and no longer indiviudally.

F. Split Reward

$$r_{noise} = I_t(G_t - L_t) \tag{18}$$

$$r_{ESS} = I_t(D_t - C_t) \tag{19}$$

$$r_{FDR} = I_t(-\sum_{p \in U_e} p_r - \sum_{p \in U_d} \frac{2p}{H-1}) \tag{20}$$

$$r_{TCL} = I_t(-\frac{1}{r_h} L_{TCL} a_{tcl,t}) \tag{21}$$

$$r_{discomfort} = -\delta exp(|T_t - \frac{T_{max} + T_{min}}{2}). \tag{22}$$

The reward for TCL was split, since maintaining the temperature is a very different task than minimizing the energy consumption.

G. Stacked Carbon Intensity Details

The carbon intensity was stacked for the next 3 hours, such that the observable became $I_t = [I_t, I_{t+1}, I_{t+2}, I_{t+3}]^H$. The resulting training curve is shown in Figure 4

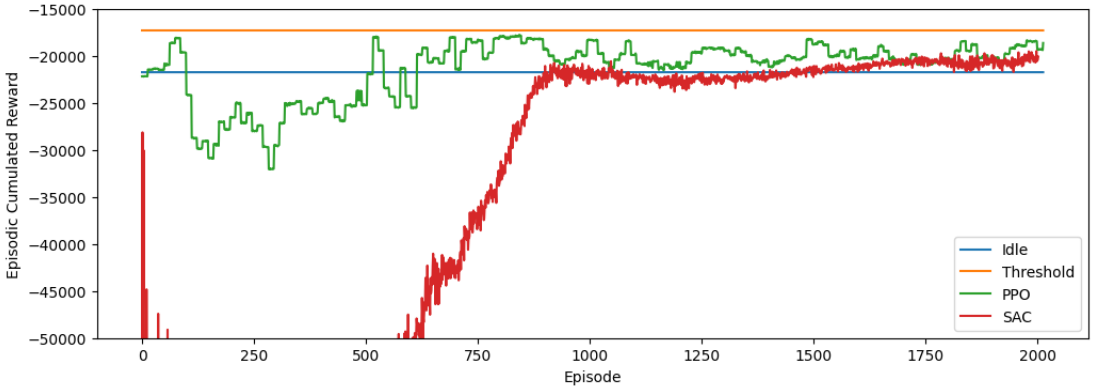

Figure 4. Training curve for the stacked carbon intensity setting.

H. Eased Exploration Details

In addition to the aforementioned changes, the FDR observable was also collapsed into the scalar only reporting the sum of lined up consumptions. The training curves for the different sub-tasks are shown in Figure 5, Figure 6, and Figure 7.

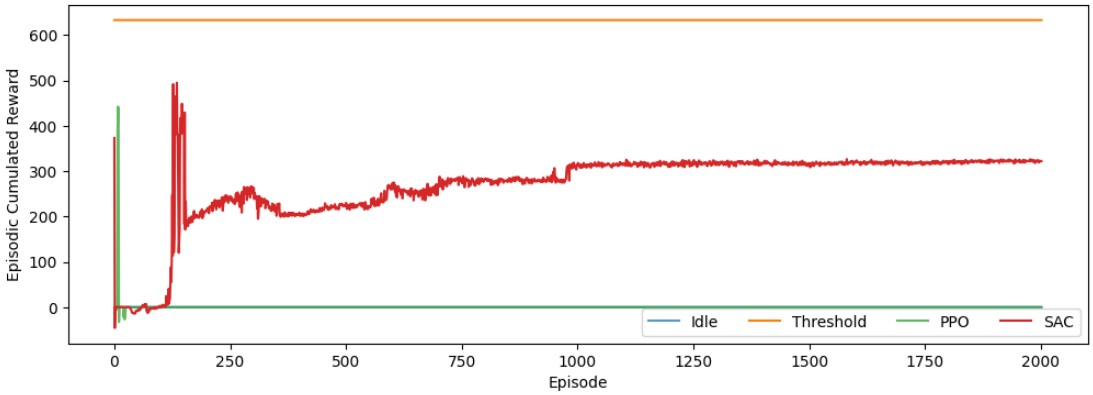

Figure 5. Training curve for ESS sub-task in the eased exploration setting.

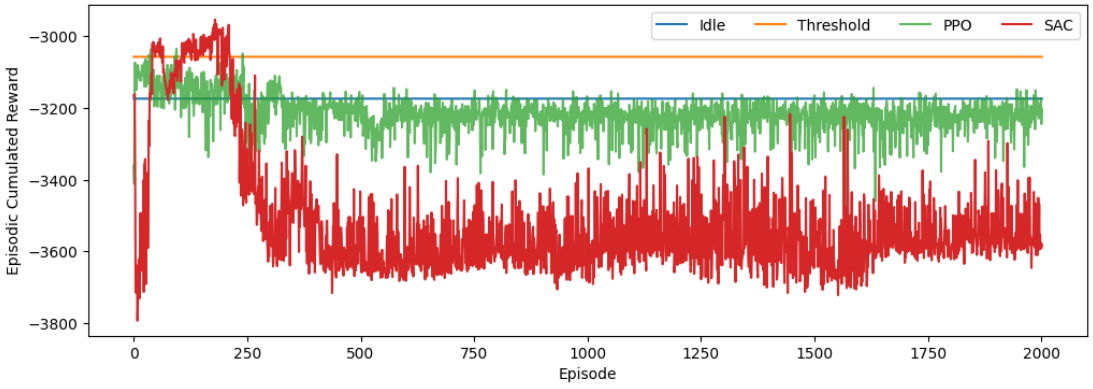

Figure 6. Training curve for FDR sub-task in the eased exploration setting.

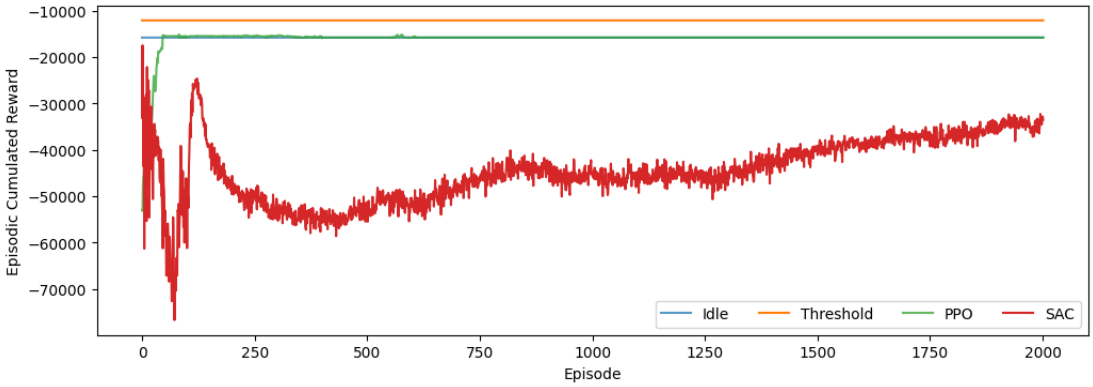

Figure 7. Training curve for TCL sub-task in the eased exploration setting.

I. Terminal Reward Details

The training curve for the terminal reward setting is shown in Figure 8.

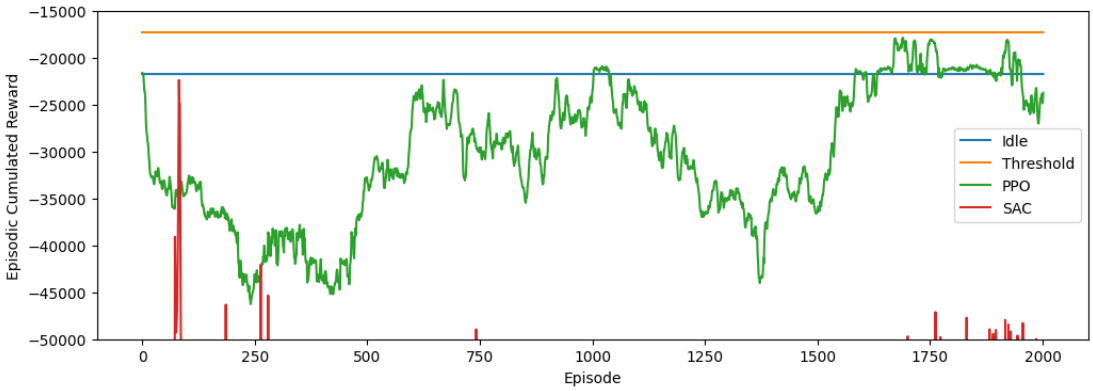

Figure 8. Training curve for the terminal reward setting.