# Deep Reinforcement Learning for Energy Management in Residential Housing

**Tim Walter** [1]

## Abstract

TODO

## 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)

Residential housing operations play a substantial role in global carbon emissions, making the optimization of energy consumption a paramount 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, addressing a significant component of worldwide emissions.

Recent global data underscores the urgency of this endeavor, with housing operations contributing a notable proportion to the overall carbon footprint. This research not only explores the reduction of emissions but also emphasizes the nuanced interplay between energy consumption and grid dynamics. Aligning energy demand with periods of high renewable output, enabled by low carbon intensity electricity, not only aids environmental goals but also enhances grid stability. Dispatching reserves through batteries during periods of high carbon intensity fosters balanced power flow, contributing to a more resilient and efficient energy grid.

Moreover, an evolving landscape in electricity contracts, featuring adaptive pricing structures, introduces a potential economic incentive for inhabitants. As the energy sector increasingly transitions towards renewable sources, synchronizing residential energy usage with favorable grid conditions not only supports ecological goals but may also translate into cost savings for residents.

This paper navigates the intricacies of these interconnected factors, shedding light on the potential of DRL as a catalyst for sustainable and economically viable energy management in residential housing.

## 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 is the 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). While MPC is a well established control method, RL is a relatively new approach that has gained a lot of attention in the last years. The main advantage of RL is that it does not require a model of the environment, which is often hard to obtain in the energy domain. However, RL is known to be sample inefficient and requires a lot of training data.

This paper is mainly inspired by (Nakabi & Toivanen, 2021), who describe a microgrid environment with generation, storage and consumers and optimized electricity pricing using various deep reinforcement learning algorithms.

## 3. Environment

The main contribution of this work is to build an environment for a single family household, consisting of a rooftop solar array for energy generation, an energy storage system, a thermostatically controlled load and flexible demand response. The enviroment is built on a mixture

[1]Department of Scientifc 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. 

of simulated and real data and dynamic components. It is used to optimize the CO2eq emissions of the household. The environment is built on the Gymansium framework (Towers et al., 2023) and is available on GitHub[1]

The training environment was chosen to be episodic to accomodate the environment framework(Towers et al., 2023) and algorithm framework(Antonin Raffin et al., 2021). The episodes are organized into weeks, with either minutely or hourly resolution, to capture the daily and weekly patterns of the household. The timeframe chosen, was 07.01.2007 - 28.12.2008, although some episodes had to be left out due to missing data. Overall there were more than 100 episodes, of which 95 were used for training, 4 for testing and the rest for training. The power was standarized to kilowatts, while the energy was simply kWmin or kWh depending on the resolution. As the agents is operating on equidistant timesteps anyway, there is no real difference between energy and power, as the power is always assumed to be constant over the timestep. The location of the house was in France near Paris.

### 3.1. Static Components

The static components are the parts of the environment that are not reactive to the agent's actions. These are auxillary information, the generation of the rooftop solar array, the household energy demand and the external electricity supply. The data was either given in hourly or minutely resolution, and was adapted to the missing resolution using linear interpolation or averaging.

#### 3.1.1. Auxillary Information

As auxillary information, weather and time data was given, which was sourced from the Photovoltaic Geographical Information System (PVGIS) (Thomas Huld et al., 2012). They had no direct influence on the reward and only provided means for the agent to predict the carbon intensity of the electricity mix. The only exception was the outdoor temperature, which influenced the thermostatically controlled load. The actual observables of this component are given in table 3.3.1.

#### 3.1.2. Rooftop Solar Array (RSA)

The generated energy was simulated using the PVLIB library (F. Holmgren et al., 2018). The simulation used the weather data described in 3.1.1 and the solar panel specifications from the CEC database (Dobos, 2012)(Boyson et al., 2007).

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

#### 3.1.3. Household Energy Demand

The household 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 modelled as inflexible demand, while the laundry room was modelled as flexible demand.

#### 3.1.4. External Electricity Supply (EES)

The direct carbon intensity of the electricity mix at any given time was sourced from electricity maps (.29, 29.12.2023) for France. However, only data from 2021 and 2022 was available, 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 gramm of $CO_2$ equivalent emissions per kWh or kWmin.

To showcase the scenario and patterns the static components are visualized for the first 24 hours in figure 1, excluding the auxillary information.

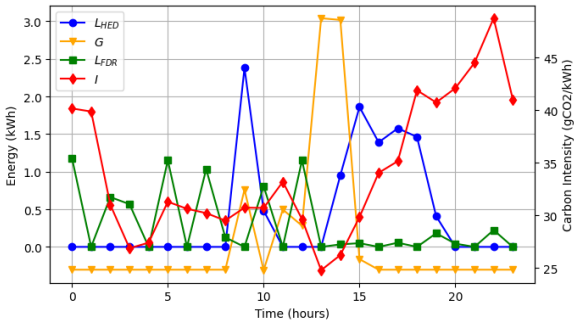


*Figure 1.* Static components for the first 24 hours

### 3.2. Dynamic Components

The dynamic components consist of an energy storage system, flexible demand response and thermostatically controlled load.

#### 3.2.1. Energy Storage System

The energy storage system (ESS) is directly chargable and dischargeable from either the EES or the RSA. The charge is subject to the followng dynamics:

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

Here, $B_t \in [0, B_{max}]$ is the charge at time $t$ and $B_{max}$ is the capacity. Furthermore, $\nu$ 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 maximum $D_{max}$. The rates can be 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 $D_t \in [0, \min\{D_{max}, (B_t - D_S)\sqrt{\nu}\}]$.

#### 3.2.2. Flexible Demand Response

Flexible Demand Response (FDR) denotes the shifting of load in time. This is modelled as a desired schedule of usage, as described in 3.1.3. A timeframe centred around the current timestep of the schedule is influenced stochatistacally by a signal $a_{fdr,t} \in [-1, 1]^H$, where $H$ is the length of the timeframe. Whether a usage is executed is determined by a Bernoulli process, with probabilities $p$ calculated from the desired schedule, the signal and an exponential weighting, which is centered around the scheduled time of usage. The probabilities are given by:

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

where $s \in \{0, 1\}^H$ is the vector of the desired schedule with elements $s_i = \begin{cases} 1 & \text{if } t_i \leq t \\ 0 & \text{else} \end{cases}$ $\beta$ is a patience parameter, and $t_s \in \mathbb{R}^H$ is the vector of desired execution times in the current timeframe. If a usage is executed its power is completely drawn in the current timestep. However, if a usage is not executed after being delayed once, it will consume a discounted amount of power in each timestep, such that if it is never exectued it will consume the same amount of power as if it was executed once. This aims to facilitate learning by preventing infinite delays or a large negative reward at the end of the episode. To do so the power consumption of a usage after the first delay is $\frac{2p}{H-1}$, which is also discounted from the remaining power.

#### 3.2.3. Thermostatically Controlled Load

The Thermostatically Controlled Load (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) that has the following update procedure:

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

where $T$ is the indoor temperature, $r_a$ is the thermal mass of the air, $T_a$ is the outdoor temperature, $r_b$ is the thermal mass of the building material, and $T_b$ is the 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 unintentional 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.

### 3.3. Markov Decision Process

The data in combination with the dynamic components allows the formulation of a Markov Decision Process (MDP), necessary for the RL paradigm. This requires the definition of an observation space, action space, a reward function and a transition function. The transition function is given by the replay of measurements described in 3.1 and the dynamics described in 3.2.

#### 3.3.1. Observation Space

The observation space is a bounded subset of $\mathbb{R}^{12}$, with the observables given in 3.3.1.

*Table 1.* Observation Space

| Observable | Symbol |
|---|---|
| Carbon Intensity | $I$ |
| Household Energy Demand | $L_{HED}$ |
| Rooftop Solar Generation | $G$ |
| ESS Charge | $B$ |
| FDR power in current timeframe | $L_{FDR}$ |
| Indoor temperature | $T$ |
| Timestep | $t$ |
| Month of Year | $MoY$ |
| Solar Irradiation | $SI$ |
| Solar Elevation | $SE$ |
| Outdor temperature | $T_a$ |
| Wind Speed | $WS$ |

#### 3.3.2. Action Space

The agent can influence the components of 3.2, through the signals $a_{fdr}$, $a_{tcl}$ and $a_{ess}$. For $a_{ess}$ the charging and discharging rates are combined into a single signal, since the ESS can either be charged or discharged at any given timestep. All signal are normalized to $[-1, 1]$ and can be

discretized into bins, if the algorithm require a discrete action space. The action space is summarized in table 3.3.2.

*Table 2.* Action Space

| ACTION | MIN. MEANING | MAX. MEANING |
|---|---|---|
| $a_{ess}$ | DISCHARGE | CHARGE |
| $a_{fdr}$ | DELAY | EXPEDITE |
| $a_{tcl}$ | NO HEATING | HEATING |

### 3.3.3. REWARD FUNCTION

The reward function is mainly given by the physical quantity of CO2eq emissions. However, to prevent unintended exploitation of the TCL, which is often optimal at constant $T_{min}$, it is augmented by a discomfort penalty. The resulting function is given by:

$$r_t = I_t(PE - CE) - DC \tag{6}$$

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

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

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

where $PE$ is the energy produced, $CE$ is the energy produced, $U_e$ is the set of executed usages, $p_r$ is the remaining power after discounting, $U_d$ is the set of delayed usages, $DC$ is the discomfort with a penalty factor $\delta$.

Moreover, since the task is episodically formulated, there is a correction necessary at the end of the episode to account for e.g. differences in charge left. The correction of the final reward is given as:

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

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

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

where $U$ is the set of left over usages. Every equation is correcting for one type of left over energy in the system.

## 4. Algorithms

To obtain a measure of success in the attempts of solving the MDP two simple baselines were implemented. The first was an idle policy, which would always return zero actions. The cumulative reward for this policy is exemplary shown for the first episode in 2. The second baseline was the following threshold policy:

$$a_t = \begin{cases} \begin{bmatrix} 1 & 1 & \psi 0.1 \end{bmatrix} & \text{if } I_t < \phi_1 \\ \begin{bmatrix} -1 & 0 & 0 \end{bmatrix} & \text{if } I_t > \phi_2 \\ \begin{bmatrix} 0 & 0 & \psi 0.1 \end{bmatrix} & \text{else} \end{cases} \tag{13}$$

where $\phi_1$ and $\phi_2$ are the lower and upper threshold respectively, while $\psi = sign(T_t - \frac{T_{max}+T_{min}}{2})$.

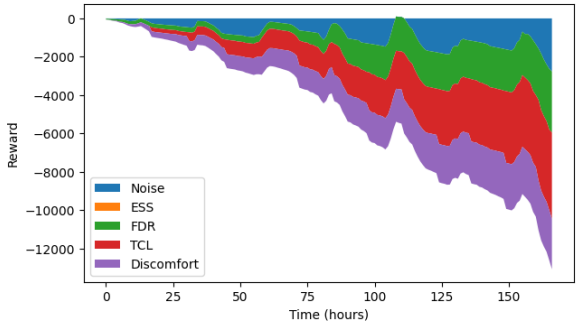


*Figure 2.* Cumulative reward for the idle policy.

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), and therefore only their core idea is explained.

SAC is a squashed Gaussian policy, that employs entropy regularization, which adds a bonus reward in each timestep 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 function for SAC is:

$$\max_{\theta} \mathop{\mathbb{E}}_{\substack{s \sim \mathcal{D} \\ \xi \sim \mathcal{N}}} \left[ \min_{j=1,2} Q_{\phi_j}(s, \tilde{a}_\theta(s, \xi)) - \alpha H(\pi_\theta \mid s) \right] \tag{14}$$

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

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

$$\theta_{k+1} = \arg\max_{\theta} \mathop{\mathbb{E}}_{\substack{s \sim \mathcal{D} \\ a \sim \pi_{\theta_k}}} [L(s, a)] \tag{15}$$

$$L(s, a) = \min \left( \left| \frac{\pi_\theta(a \mid s)}{\pi_{\theta_k}(a \mid s)} \right|, 1 \pm \epsilon \right) A^{\pi_{\theta_k}}(s, a) \tag{16}$$

where $A^{\pi_{\theta_k}}(s, a)$ is the advantage of the current policy, and $\epsilon$ the trust region hyperparameter, clipping the update

size. Alternatively, PPO can also be implemented using a KL-divergence constraint.

## 5. Experiments

To investigate the ease of the given problem initially, the first experiments were replaying only a single epsiode. Furthermore, the initial conditions, the initial indoor temperature $T_0$, building mass temperature $T_{b,0}$, and initial charge $B_0$, were deterministic for every training episode, but would be shuffled later.

Since the task can be split nicely into distinct subtasks for each dynamic component, the rewards were tracked separatly to allow the analysis of performance of each subtask explicitly. This split the reward into the following components:

$$r_{noise} = I_t(G_t - L_t) \tag{17}$$

$$r_{ESS} = I_t(D_t - C_t) \tag{18}$$

$$r_{FDR} = I_t(-\sum_{p \in U_e} p_r - \sum_{p \in U_d} \frac{2p}{H-1}) \tag{19}$$

$$r_{TCL} = I_t(-\frac{1}{r_h} L_{TCL} a_{tcl,t}) \tag{20}$$

$$r_{discomfort} = -\delta exp(|T_t - \frac{T_{max} + T_{min}}{2}) \tag{21}$$

The reward for TCL was split, since maintaining the temperature is a very different task than minimizing the energy consumption. The result of all experiments are shown in 5.

Unfortunately, to date the task has not been solved successfully, as the RL algorithms are not outperforming the baselines. Since the algorithms were widely tested and used in other domains, the problem is likely to be in the formulation of the problem. Therefore, three hypothesis are formulated to explain the poor performance of the algorithms and attempts to solve them are discussed in the following. The experiments were run with cumultating changes. The training details can be found in A.

### 5.1. Hypothesis 1: Scheduling, Stochasticity, and Space Dimensionality

The first issue investigated, was the FDR formulation. The problem of scheduling using RL is already challenging (Zhang et al.) but is further complicated since the agent has only stochastic control. Furthermore, the action space and observation space dimensionality was extremely high, especially in minutely resolution, since the desired planning horizon was 24 hours. This could lead to a curse of dimensionality (Sutton & Barto, 2018) and was especially poor since the dimensionality would depend on the parametrization of the environment both in terms of $H$ and the resolution.

The attempted solution was to simplify the task, by allowing the agent to control the FDR deterministically and only give a scalar signal for all components. These changes are equivalent to the assumption that the optimal policy is invariant to $U_e$ as long as the sum is maintained and that $\beta = \infty$.

### 5.2. Hypothesis 2: Hard Exploration

Another difficulty of the environment, is its vast exploration space, given by the continuity of the action and observation space, the length of the episodes, especially in minutely resolution, and the fact that the agent has to perform multiple actions per timestep. The task is furter complicated by the noise from the RSA and HED on the reward. Moreover, a lot of observables are only relevant for estimating the future carbon intensity.

To address these concerns the minutely resolution was omitted from now on. In addition, the action space for PPO was discretized into 11 bins for each component, the HED and RSA were removed from the observables and reward calculation, and the policies were trained separatly for every subtask. Moreover, observables that did not impact the reward directly besides, the timestep were removed. Lastly, the replay buffer for SAC was initialized with one episode of transitions of the threshold algorithm, providing a warm start (Wang et al.).

### 5.3. Hypothesis 3: Delayed or Deceptive reward

Currently, the last idea to explain the poor performance of the algorithms 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, which can be problematic (Sutton, 1984). Moreover, the rewards might even be deceptive, since those actions are necessary for an effective policy but only ever receive negative reward.

To address the time missmatch between action and reward, the reward was accumulating over time and given as an observable, however only actuated as a reward in the terminal state of the episode, which should encourage the agent to evaluate the entire trajectory of an episode together.

## 6. Discussion

## References

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

*Table 3.* Accumulated rewards split by origin per experiment

| Experiment | Algorithm | ESS | FDR | TCL | Discomfort |
|---|---|---|---|---|---|
| Full Environment | Idle | 0.00 | 0.00 | 0.00 | 0.00 |
| | Threshold | 0.00 | 0.00 | 0.00 | 0.00 |
| | PPO | 0.00 | 0.00 | 0.00 | 0.00 |
| | SAC | 0.00 | 0.00 | 0.00 | 0.00 |
| Hypothesis 1 | Idle | 0.00 | 0.00 | 0.00 | 0.00 |
| | Threshold | 0.00 | 0.00 | 0.00 | 0.00 |
| | PPO | 0.00 | 0.00 | 0.00 | 0.00 |
| | SAC | 0.00 | 0.00 | 0.00 | 0.00 |
| Hypothesis 2 | Idle | 0.00 | 0.00 | 0.00 | 0.00 |
| | Threshold | 0.00 | 0.00 | 0.00 | 0.00 |
| | PPO | 0.00 | 0.00 | 0.00 | 0.00 |
| | SAC | 0.00 | 0.00 | 0.00 | 0.00 |
| Hypothesis 3 | Idle | 0.00 | 0.00 | 0.00 | 0.00 |
| | Threshold | 0.00 | 0.00 | 0.00 | 0.00 |
| | PPO | 0.00 | 0.00 | 0.00 | 0.00 |
| | SAC | 0.00 | 0.00 | 0.00 | 0.00 |


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.

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.

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.

Sutton, R. S. *Temporal Credit Assignment in Reinforcement Learning*. PhD thesis, 1984.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. URL http://incompleteideas.net/book/the-book-2nd.html.

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.

Wang, H., Lin, S., and Zhang, J. Warm-start actor-critic: From approximation error to sub-optimality gap. URL http://arxiv.org/pdf/2306.11271.pdf.

Zhang, C., Song, W., Cao, Z., Zhang, J., Tan, P. S., and Xu, C. Learning to dispatch for job shop scheduling via deep reinforcement learning. URL http://arxiv.org/pdf/2010.12367.pdf.

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. Training Procedure

All experiments were run for 250 episodes and in hourly resolution. The training parameters are shown in A.

*Table 4.* Training Parameters

| PARAMETER | VALUE |
|---|---|
| $B_{max}$ | 13.5 |
| $C_{max}$ | 1 |
| $D_{max}$ | 1 |
| $\nu$ | 0.95 |
| $D_s$ | 0.01 |
| $B_0$ | 0 |
| $H$ | 25 |
| $\beta$ | 20 |
| $\delta$ | 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 |