Addressing partial observability in reinforcement learning for energy management

Marco Biemann*
Technical University of Denmark
Department of Technology, Management and Economics
Kgs. Lyngby, Denmark
marcob@dtu.dk

Xiufeng Liu
Technical University of Denmark
Department of Technology, Management and Economics
Kgs. Lyngby, Denmark
xiuli@dtu.dk

Yifeng Zeng
Northumbria University
Department of Computer and Information Sciences
Newcastle upon Tyne, United Kingdom
yifeng.zeng@northumbria.ac.uk

Lizhen Huang
Norwegian University of Science and Technology
Department of Manufacturing and Civil Engineering
Gjøvik, Norway
lizhen.huang@ntnu.no

ABSTRACT

Automatic control of energy systems is affected by the uncertainties of multiple factors, including weather, prices and human activities. The literature relies on Markov-based control, taking only into account the current state. This impacts control performance, as previous states give additional context for decision making. We present two ways to learn non-Markovian policies, based on recurrent neural networks and variational inference. We evaluate the methods on a simulated data centre HVAC control task. The results show that the off-policy stochastic latent actor-critic algorithm can maintain the temperature in the predefined range within three months of training without prior knowledge while reducing energy consumption compared to Markovian policies by more than 5%.

CCS CONCEPTS

• **Computing methodologies** → **Reinforcement learning**; • **Mathematics of computing** → *Markov processes*; **Variational methods**.

KEYWORDS

Reinforcement learning, HVAC control, energy management, POMDP, recurrent neural networks, variational inference.

ACM Reference Format:
Marco Biemann, Xiufeng Liu, Yifeng Zeng, and Lizhen Huang. 2021. Addressing partial observability in reinforcement learning for energy management. In *The 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '21), November 17–18, 2021, Coimbra, Portugal.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3486611.3488730

*Also with Norwegian University of Science and Technology.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
BuildSys '21, November 17–18, 2021, Coimbra, Portugal
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-9114-6/21/11…$15.00
https://doi.org/10.1145/3486611.3488730

1 INTRODUCTION

Smart control in the energy sector is a complex process whose control decisions are influenced by exogenous signals in the form of time-series data, such as ambient temperature, solar radiation, energy prices and energy demand. However, these signals are notoriously difficult to understand and predict, mainly due to their high variance. In recent years, reinforcement learning (RL) in automatic control systems has gained significant attention. RL does not explicitly model these uncertainties but instead learns from the interaction with the environment. Most existing controllers assume the Markov assumption, i.e., the decision making depends only on the current state (not on past states). This assumption is reasonable for chess or Go or for the robot locomotion tasks in OpenAI Gym [5], as their state space consists of positions and velocities of the joints, determining the environment completely. In most real-world control scenarios, the Markov assumption is however inaccurate. To address this, past information is commonly used for decision-making. For image-based observations, Mnih et al. [25] use frame stacking to infer the direction and velocity of moving objects.

In practice, the measurements of the environment are often insufficient to model the system precisely. This applies especially to thermal control applications, as modelling non-equilibrium thermodynamics is challenging. For example, in a data centre, the state information depends on various factors, including CPU load, temperatures of the servers and building materials, dynamics of the airflows in the room, and more. It is not realistic to measure all of these factors, and good approximations are costly. Privacy is an important concern for centralised implementations of multi-agent systems. In these scenarios, only partial information is available, and how to use this limited information for control becomes a challenge.

Such problems can be formulated as a *Partially Observable Markov Decision Process* (POMDP), which is a generalisation of MDP. This formulation can be beneficial for two reasons. History-dependent policies are helpful, as they can recognise the recurrent patterns in the exogenous time-series data. Further, by explicitly formulating that the state is unknown, the agent can learn a latent state representation, which can benefit learning in non-stationary environments.

The idea of maintaining a belief state in partially observable environments in optimal control is due to Åström [1]. It has been studied in the control literature [3, 32] and was revisited recently in RL using variational inference [17]. Another line of work that does not rely on learning explicitly the state rely on recurrent neural networks (RNN) and especially LSTM [15], an architecture addressing vanishing gradients. LSTM was first applied to RL for solving tasks requiring long-term memory [2, 37] and plays a central role in various major achievements of RL in games [18, 27, 35]. Regarding the energy sector, Ruelens et al. [31] highlighted that typical environments in energy management are not fully observable and encoded past observations into an autoencoder. Wang et al. [36] used LSTM in an RL actor-critic algorithm, whereas Zhang et al. [39] used LSTM in a model-based RL algorithm to learn environmental dynamics. Sequence-to-sequence models [6–8] and Bayesian networks [16, 28] were applied to make predictions in model predictive control. Soft actor-critic (SAC) [10] is chosen in this work due to its promising results in energy management [4, 21, 30, 40].

In this paper, we present two approaches based on POMDPs on a simulated HVAC control case study. The first approach changes the network architectures to gated RNNs, taking sequences of past observations as input. The second one consists of inferring the state by learning the belief with variational inference. Both methods demonstrate their effectiveness by obtaining state-of-the-art results in terms of data and energy efficiency.

2 PROBLEM FORMULATION

A *Partially Observable Markov Decision Process* (POMDP) is a generalisation of an MDP. A POMDP is a septuple $(\mathcal{S}, \mathcal{A}, \mathcal{O}, p, e, \rho, r)$, where:

- $\mathcal{S}$ is the *state space*, that is all the sufficient and necessary information to model the transitions and rewards;
- $\mathcal{A}$ is the *action space*;
- $\mathcal{O}$ is the *observation space*, corresponding to the measurements available to the agent;
- p are the *state-transition probabilities* of going from state $s \in \mathcal{S}$ to state $s' \in \mathcal{S}$ using action $a \in \mathcal{A}$;
- e are the *emission probabilities* of observing (or measuring) observation $o \in \mathcal{O}$ in state $s \in \mathcal{S}$;
- ρ is the *intial state probability* of starting at state $s \in \mathcal{S}$;
- $r : \mathcal{S} \times \mathcal{A} \longrightarrow \mathbb{R}$ is the reward function.

The POMDP model reduces to an MDP if $\mathcal{S} = \mathcal{O}$. We assume that all probabilities are unknown to the agent. We define the *history* $h_t = (o_0, a_0, \cdots, a_{t-1}, o_t)$ as the available information at timestep t. We define the probability of trajectories $\tau = (s_t, o_t, a_t)_{t \geq 0}$ by:

$$p_\pi(d\tau) = \rho(ds_0) \prod_{t \geq 0} e(do_t \mid s_t) \pi(da_t \mid h_t) p(ds_{t+1} \mid a_t, s_t)$$

and aim to find a probability distribution π^* that maximises for $\gamma \in (0, 1)$ the following objective:

$$J(\pi) = \mathbb{E}_{\tau \sim p_\pi} \left[\sum_{t \geq 0} \gamma^t r(s_t, a_t) \right].$$

Compared with MDP, the policy is conditioned on the whole history h_t, instead of the last observation o_t. That implies that the Q-function $Q_\pi(h_t, a_t)$ needs to be defined over the whole history as well. This is problematic as its dimension grows over time.

3 METHODS

This section will address the above research problem by presenting two distinct approaches, relying on actor-critic methods. The first approach assumes that RNNs parameterise the actor and critic to deal with sequential data. This change can be straightforwardly implemented into standard algorithms, such as Proximal Policy Optimisation (PPO) [33]. The second approach aims to learn a latent representation of the state, that can be used by the actor and critic, instead of the whole history. It starts with an uninformative prior $b_0(s_0)$ and updates the belief $b_t(s_t \mid h_t)$ with new evidence using Bayesian statistics. It is a classical method that has been widely studied, e.g., [1, 20, 32, 34]. This method can lead to more robust policies in non-stationary environments [38] and more explainable results.

3.1 Recurrent neural networks

Standard implementations of actor-critc methods, such as PPO, use separate feed-forward networks for the actor and the critic. Following Jaderberg et al. [19], the architecture is modified so that observations go through an LSTM cell instead. Past observations are encoded into a recurrent state, that is shared between the actor and critic. The long-term memory c_t is beneficial to identify the recurrent patterns of weather data. The advantage of this approach is that the RL algorithm does not have to be modified, except for the architecture and the shorter time horizon, to avoid backpropagating too far through time. Figure 1 describes the neural network architecture of PPO with LSTM used in StableBaselines [14].

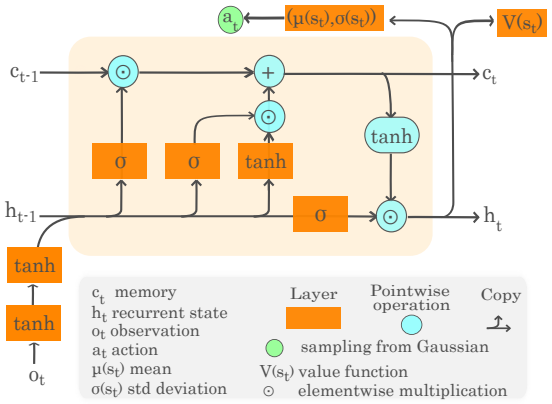
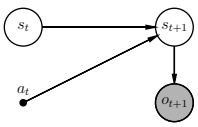


Figure 1: Neural network architecture of PPO with LSTM.

Off-policy methods typically learn the Q-function. The architecture, described in Figure 1 can only be used for the actor; the critic needs to be conditioned on actions as well. Off-policy methods typically use an architecture similar to [13, 29]. A notable difference is that we need to store short trajectories into the replay buffer instead of simple transitions. However, a major concern of LSTM for off-policy methods is its high computational cost, as typically three LSTMs need to be trained [9] for better performance.

3.2 Learning the belief

A *belief* b is a probability measure in $\mathcal{S}$ over states. Given a prior belief b_t and action a_t, we compute the posterior $b_{t+1}(s_{t+1} \mid b_t, a_t, o_{t+1})$, using the new observation o_{t+1}. This can be expressed by the following graphical model:



By applying Bayes' theorem, the belief can, in theory, be updated recursively:

$$b_{t+1}(s_{t+1} \mid b_t, a_t, o_{t+1}) \propto \int_{\mathcal{S}} p(s_{t+1} \mid a_t, s_t) e(o_{t+1} \mid s_{t+1}) b_t(ds_t). \tag{1}$$

By updating the belief this way, we can show that it is a sufficient statistic [3], meaning that the belief summarises all the information from the past required for decision making. In particular, future rewards can be estimated by $Q_\pi(s_t, a_t)$ with $s_t \sim b_t$.

However, the update (1) is intractable as it requires knowledge of the model. Therefore, the belief needs to be updated in an approximate way, which can be done with variational autoencoders (VAE) [22]. A possible way to solve this is the stochastic latent actor-critic (SLAC) algorithm [23], that is a natural generalisation of SAC to POMDPs. Other studies combine VAE with RNNs [12, 17].

We restrict the beliefs to Gaussian distributions, as a posterior of a Gaussian remains Gaussian. The distribution q_φ is an inference model (or encoder), that aims to find a latent representation of the state, given the observed data. The encoder is a neural network updating the state using the most recent data (a_{t-1}, o_t):

$$s_t \sim q_\varphi(\cdot \mid s_{t-1}, a_{t-1}, o_t). \tag{2}$$

The state s_t is what we are interested in, but as we cannot solve (1) exactly, we should ensure that q_φ is a sufficient statistic. That is, the representation s_t needs to encode information generating the observations o_t given by the environment and be updated when an action is taken, so that the next state s_{t+1} explains the next observation o_{t+1}. Therefore, we train a generative model (or decoder) consisting of two networks:

$$\hat{o}_t \sim e_\psi(\cdot \mid s_t),$$
$$\hat{s}_{t+1} \sim p_\psi(\cdot \mid s_t, a_t),$$

where we denote by a hat the observations and states generated by the decoder. As (2) is conditioned on the previous state, we construct a Bayesian network (where $s_0 \sim q_\varphi(\cdot \mid o_0)$ and $\hat{s}_0 \sim \mathcal{N}(0, I)$) to generate a sequence of states $(s_0, \cdots, s_T)$. The networks are updated in order to:

- maximise the likelihood of observations $\sum_{t=0}^{T} \log e_\psi(\cdot \mid s_t)$,
- minimise $\sum_{t=0}^{T} \mathcal{D}_{KL}\left(q_\varphi(\cdot \mid s_{t-1}, a_{t-1}, o_t) \| p_\psi(\cdot \mid s_{t-1}, a_{t-1})\right)$.

The actor $\pi_\theta(a_t \mid s_t)$ and the critics $Q_w(s_t, a_t)$ can be defined and updated the same way as for SAC, conditioned on $s_t \sim q_\varphi$. This approach can also be applied to other off-policy methods. The design used by Lee et al. [23] (and in the experiments) is more complicated and uses a latent variable factorisation for better performance.

SLAC is strongly related to model-based algorithms, such as Dreamer [11]. However, a significant difference is that in SLAC, the environment model is only used in the loss function to infer better states, used by the critic to predict rewards. The predictions of the model are not used by the policy to make decisions or as additional training data.

4 EXPERIMENT

We evaluate the methods based on a classical HVAC control case study simulated with EnergyPlus, whose RL environment was implemented by Moriyama et al. [26] and has been used in [4, 24, 39]. It represents a two-zone medium-sized data centre, whose objective is to reduce energy consumption, while maintaining the indoor temperatures within a predefined range. The observation space consists of the outdoor air temperature, the indoor temperature in both zones and the electricity demand of the servers P_{it} (in kW) and HVAC system P_{hvac}. The actions consist of changing the temperature setpoints of the HVAC system and adjusting the airflow rate in both zones. We used the following reward function:

$$r(s, a) = R_{\text{west}} + R_{\text{east}} - \lambda_P \left(P_{\text{it}} + P_{\text{hvac}}\right),$$

where R_i is the reward obtained when maintaining temperature in zone i in the range. The term R_i is defined as:

$$R_i = \exp\left(-\lambda_1 \left(T_i - T_{\text{tgt}}\right)^2\right) - \lambda_2 \left([T_{\text{min}} - T_i]_+ + [T_i - T_{\text{max}}]_+\right), \tag{3}$$

where $[\text{T}_{\text{min}}, \text{T}_{\text{max}}]$ is the desired range, T_{tgt} is the midpoint of the interval and $[x]_+ = \max(x, 0)$. We used $\text{T}_{\text{min}} = 23\ ^\circ C$, $\text{T}_{\text{max}} = 24^\circ C$, $\lambda_P = 10^{-5}$, $\lambda_1 = 0.5$, $\lambda_2 = 0.1$. As in [26], we used a tighter range in the reward function for better temperature control to further insure that the temperatures lie between $22^\circ C$ and $25^\circ C$. The first term in (3) corresponds to a Gaussian, centered at the desired temperature; the second corresponds to a trapezoid, helping training when the temperature is far away from the target, as the Gaussian would tend too quickly to 0. For more details about the case study, we refer to [4, 26].

The classical algorithms, PPO, PPO-LSTM and SAC, are implemented with the Stable Baselines framework [14] and its original hyperparameters. For SAC-LSTM, we use the architecture from [29]. As the original SLAC architecture is implemented for image observations, it has to be modified for the current case study, following the implementation by Han et al. [12][1]. Given their large influence on the performance of algorithms, we provide details about the used architectures and hyperparameters in supplementary material on our website[2]. We also present there additional figures and results about the experiments done in Section 5.

5 RESULTS AND ANALYSIS

Our discussion will focus on energy consumption and data-efficiency. The ability to maintain the temperature within the desired range

[1]The implementation is available at https://github.com/oist-cnru/Variational-Recurrent-Models. The repository contains implementations of SAC-LSTM, SLAC and their own algorithm SAC-VRM.
[2]https://biemann.github.io/rlem2021

is essential, but all algorithms (except PPO with a feed-forward network) can handle this task within 20 years of training, obtaining similar results (although we observed that SLAC is especially good at this task).

In Figure 2, we compare the power consumption for all algorithms. The algorithms are trained without any prior knowledge. We observe that the models specialised for POMDPs (SLAC, SAC-LSTM, PPO-LSTM) can significantly reduce consumption and outperform the baseline controller implemented into EnergyPlus within one month. The algorithms show similar improvements in terms of temperature control, as shown in Figure 3 for SLAC, and the temperatures lie predominantly in the range after three months. We observed similar results for SAC-LSTM (SAC takes around one year). PPO-LSTM takes a few years until it manages to maintain the temperatures in the range (see Figure 3 for the first episode). It still increases data-efficiency considerably, compared to traditional PPO.

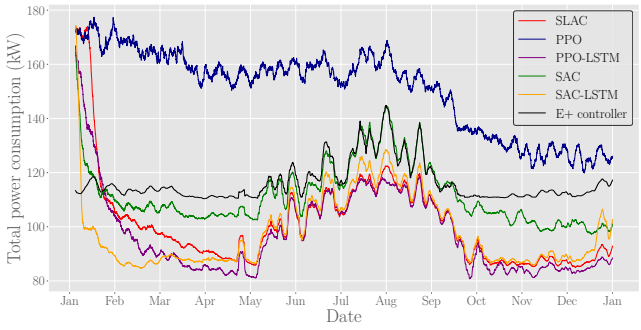


Figure 2: Comparison of algorithms during the first episode

After 20 years of training, we observe in Figure 4 that PPO-LSTM consistently has the lowest consumption, reducing it relative to traditional PPO by 7.8% (and by 20.0% relative to the baseline). SLAC also consistently outperforms the conventional RL algorithms, reducing energy consumption by 5.2% compared to SAC and by 15.3% compared to the baseline. SAC and SAC-LSTM have similar performance in terms of consumption. We observed that SAC-LSTM is better at maintaining temperatures than SAC, but sensitive to hyperparameters and prone to catastrophic forgetting. We found that the other algorithms are robust. This observation and the lower energy consumption suggest that it may be preferable to choose SLAC over SAC-LSTM as a choice of non-Markovian off-policy algorithm.

The significant improvements of SLAC over SAC and PPO-LSTM over PPO in terms of energy consumption, temperature management and data-efficiency suggest that policies that can remember past information may be helpful in stochastic environments. The use of non-Markovian policies can give new insights into the choice between on-policy (PPO) and off-policy (SAC, SLAC) methods, for instance, discussed by [4]. Off-policy methods remain more data-efficient and can reach a good policy quickly, but show only minor improvements after a few episodes. In contrast, PPO-LSTM achieves similar temperature stability after a few episodes while reducing energy consumption significantly.

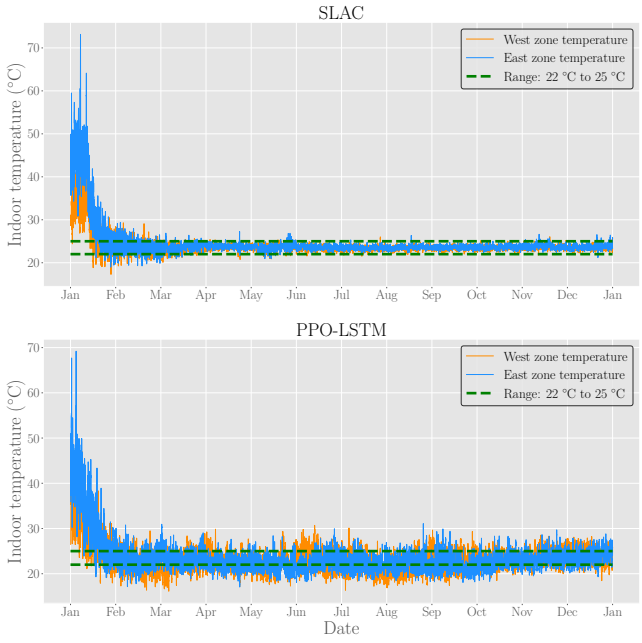


Figure 3: First training episode of SLAC and PPO-LSTM.

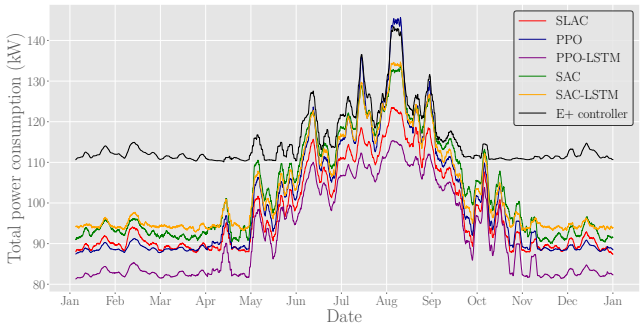


Figure 4: Comparison of algorithms on test location.

6 CONCLUSION

Non-Markovian policies can identify tendencies in the recent past, such as whether temperatures increased in recent hours. This additional insight can allow policies to outperform Markovian policies, suggesting that the formulation of RL in terms of MDP is inaccurate for energy management applications. We found that the SLAC algorithm is more data-efficient and reduces energy consumption, compared to SAC and the baseline respectively. Similarly, the use of an LSTM can improve the results of PPO, and reduce energy consumption significantly. The data-efficiency of SLAC, combined with imitation learning, should close the gap towards training an RL controller directly in the real world.

Future work should investigate whether non-Markovian policies can achieve competitive results with policies using weather or price forecasts as input. An extension of these methods to model-based RL algorithms is natural, as they are based on similar concepts.

REFERENCES

[1] Karl J Astrom. 1965. Optimal control of Markov decision processes with incomplete state estimation. *J. Math. Anal. Applic.* 10 (1965), 174–205.

[2] Bram Bakker. 2001. Reinforcement Learning with Long Short-Term Memory. In *NIPS*.

[3] Dimitri P Bertsekas and Steven E Shreve. 1996. *Stochastic optimal control: the discrete-time case*. Vol. 5. Athena Scientific.

[4] Marco Biemann, Fabian Scheller, Xiufeng Liu, and Lizhen Huang. 2021. Experimental evaluation of model-free reinforcement learning algorithms for continuous HVAC control. *Applied Energy* 298 (2021), 117164.

[5] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. OpenAI Gym. arXiv:arXiv:1606.01540

[6] Bingqing Chen, Weiran Yao, Jonathan Francis, and Mario Bergés. 2020. Learning a distributed control scheme for demand flexibility in thermostatically controlled loads. In *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE, 1–7.

[7] Yize Chen, Yuanyuan Shi, and Baosen Zhang. 2017. Modeling and optimization of complex building energy systems with deep neural networks. In *2017 51st Asilomar Conference on Signals, Systems, and Computers*. IEEE, 1368–1373.

[8] Matthew J Ellis and Venkatesh Chinde. 2020. An encoder–decoder LSTM-based EMPC framework applied to a building HVAC system. *Chemical Engineering Research and Design* 160 (2020), 508–520.

[9] Scott Fujimoto, Herke van Hoof, and David Meger. 2018. Addressing function approximation error in actor-critic methods. *Proceedings of Machine Learning Research* 80 (2018), 1587–1596.

[10] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *International Conference on Machine Learning*. 1861–1870.

[11] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. 2019. Dream to Control: Learning Behaviors by Latent Imagination. In *International Conference on Learning Representations*.

[12] Dongqi Han, Kenji Doya, and Jun Tani. 2020. Variational Recurrent Models for Solving Partially Observable Control Tasks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=r1lL4a4tDB

[13] Nicolas Heess, Jonathan J Hunt, Timothy P Lillicrap, and David Silver. 2015. Memory-based control with recurrent neural networks. *arXiv preprint arXiv:1512.04455* (2015).

[14] Ashley Hill, Antonin Raffin, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, Rene Traore, Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, and Yuhuai Wu. 2018. Stable Baselines. https://github.com/hill-a/stable-baselines.

[15] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[16] Md Monir Hossain, Tianyu Zhang, and Omid Ardakanian. 2021. Identifying grey-box thermal models with Bayesian neural networks. *Energy and Buildings* 238 (2021), 110836.

[17] Maximilian Igl, Luisa Zintgraf, Tuan Anh Le, Frank Wood, and Shimon Whiteson. 2018. Deep variational reinforcement learning for POMDPs. In *International Conference on Machine Learning*. PMLR, 2117–2126.

[18] Max Jaderberg, Wojciech M Czarnecki, Iain Dunning, Luke Marris, Guy Lever, Antonio Garcia Castaneda, Charles Beattie, Neil C Rabinowitz, Ari S Morcos, Avraham Ruderman, et al. 2019. Human-level performance in 3D multiplayer games with population-based reinforcement learning. *Science* 364, 6443 (2019), 859–865.

[19] Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z. Leibo, David Silver, and Koray Kavukcuoglu. 2017. Reinforcement Learning with Unsupervised Auxiliary Tasks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. https://openreview.net/forum?id=SJ6yPD5xg

[20] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. 1998. Planning and acting in partially observable stochastic domains. *Artificial intelligence* 101, 1-2 (1998), 99–134.

[21] Anjukan Kathirgamanathan, Eleni Mangina, and Donal P. Finn. 2021. Development of a Soft Actor Critic deep reinforcement learning approach for harnessing energy flexibility in a Large Office building. *Energy and AI* 5 (2021), 100101. https://doi.org/10.1016/j.egyai.2021.100101

[22] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. http://arxiv.org/abs/1312.6114

[23] Alex Lee, Anusha Nagabandi, Pieter Abbeel, and Sergey Levine. 2020. Stochastic Latent Actor-Critic: Deep Reinforcement Learning with a Latent Variable Model. *Advances in Neural Information Processing Systems* 33 (2020).

[24] Yuanlong Li, Yonggang Wen, Dacheng Tao, and Kyle Guan. 2019. Transforming cooling optimization for green data center via deep reinforcement learning. *IEEE transactions on cybernetics* 50, 5 (2019), 2002–2013. https://doi.org/10.1109/tcyb.2019.2927410

[25] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature* 518, 7540 (2015), 529–533.

[26] Takao Moriyama, Giovanni De Magistris, Michiaki Tatsubori, Tu-Hoa Pham, Asim Munawar, and Ryuki Tachibana. 2018. Reinforcement Learning Testbed for Power-Consumption Optimization. In *Methods and Applications for Modeling and Simulation of Complex Systems*. Springer Singapore, Singapore, 45–59.

[27] OpenAI, Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique Pondé de Oliveira Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. 2019. Dota 2 with Large Scale Deep Reinforcement Learning. (2019). arXiv:1912.06680 https://arxiv.org/abs/1912.06680

[28] Nilavra Pathak, James Foulds, Nirmalya Roy, Nilanjan Banerjee, and Ryan Robucci. 2019. A Bayesian Data Analytics Approach to Buildings' Thermal Parameter Estimation. In *Proceedings of the Tenth ACM International Conference on Future Energy Systems*. 89–99.

[29] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. 2018. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 3803–3810.

[30] Giuseppe Pinto, Marco Savino Piscitelli, José Ramón Vázquez-Canteli, Zoltán Nagy, and Alfonso Capozzoli. 2021. Coordinated energy management for a cluster of buildings through deep reinforcement learning. *Energy* 229 (2021), 120725.

[31] Frederik Ruelens, Sandro Iacovella, Bert J Claessens, and Ronnie Belmans. 2015. Learning agent for a heat-pump thermostat with a set-back strategy using model-free reinforcement learning. *Energies* 8, 8 (2015), 8300–8318. https://doi.org/10.3390/en8088300

[32] Stuart Russell and Peter Norvig. 2002. Artificial intelligence: a modern approach. (2002).

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

[34] Richard D Smallwood and Edward J Sondik. 1973. The optimal control of partially observable Markov processes over a finite horizon. *Operations research* 21, 5 (1973), 1071–1088.

[35] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* 575, 7782 (2019), 350–354.

[36] Yuan Wang, Kirubakaran Velswamy, and Biao Huang. 2017. A long-short term memory recurrent neural network based reinforcement learning controller for office heating ventilation and air conditioning systems. *Processes* 5, 3 (2017), 46. https://doi.org/10.3390/pr5030046

[37] Daan Wierstra, Alexander Förster, Jan Peters, and Jürgen Schmidhuber. 2010. Recurrent policy gradients. *Logic Journal of the IGPL* 18, 5 (2010), 620–634.

[38] Annie Xie, James Harrison, and Chelsea Finn. 2021. Deep Reinforcement Learning amidst Continual Structured Non-Stationarity. In *International Conference on Machine Learning*. PMLR, 11393–11403.

[39] Chi Zhang, Sanmukh R Kuppannagari, Rajgopal Kannan, and Viktor K Prasanna. 2019. Building HVAC scheduling using reinforcement learning via neural network based model approximation. In *Proceedings of the 6th ACM international conference on systems for energy-efficient buildings, cities, and transportation*. 287–296. https://doi.org/10.1145/3360322.3360861

[40] Tianyu Zhang, Gaby Baasch, Omid Ardakanian, and Ralph Evins. 2021. On the Joint Control of Multiple Building Systems with Reinforcement Learning. In *Proceedings of the Twelfth ACM International Conference on Future Energy Systems*. 60–72. https://doi.org/10.1145/3447555.3464855