Explainable Home Energy Management Systems based on Reinforcement Learning using Differentiable Decision Trees

Gargya Gokhale
gargya.gokhale@ugent.be
IDLab, Ghent University — imec
Ghent, Belgium

Bert Claessens
IDLab, Ghent University — imec
Beebop.ai
Belgium

Chris Develder
IDLab, Ghent University — imec
Ghent, Belgium

ABSTRACT

With the ongoing energy transition, demand-side flexibility has become important to provide grid support and allow further integration of sustainable energy sources. Residential energy assets constitute a major and largely untapped source of flexibility, driven by the increased adoption of solar PV, home batteries, and EVs. However, unlocking this residential flexibility is challenging, as it requires a control framework that can effectively manage household energy consumption while maintaining user comfort, which should be easily scalable across different, diverse houses. We aim to address this challenging problem and introduce a reinforcement learning-based approach using differentiable decision trees. Our proposed approach integrates the scalability of data-driven reinforcement learning with the explainability of (differentiable) decision trees. The resulting controller can be easily adapted across different houses and provides a simple control policy that can be explained to end-users, facilitating maximal user acceptance. As a proof-of-concept, we analyze our method using a home energy management problem, comparing its performance with commercially available rule-based baseline and conventional state-of-the-art neural network-based RL controllers. Our preliminary study indicates that the proposed method performs comparable to standard RL-based controllers, outperforming baseline controllers by ∼20% in terms of daily cost savings while being straightforward to explain.

CCS CONCEPTS

• **Theory of computation → Reinforcement learning**; • **Hardware → Smart grid**; • **Computing methodologies → Rule learning**.

KEYWORDS

Reinforcement Learning, Explainable AI, Home Energy Management, Differentiable decision trees, Demand Response

ACM Reference Format:
Gargya Gokhale, Bert Claessens, and Chris Develder. 2024. Explainable Home Energy Management Systems based on Reinforcement Learning using Differentiable Decision Trees. In *The 15th ACM International Conference on Future and Sustainable Energy Systems (E-Energy '24), June 04–07, 2024, Singapore, Singapore.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3632775.3663310

E-Energy '24, June 04–07, 2024, Singapore, Singapore

ACM ISBN 979-8-4007-0480-2/24/06
https://doi.org/10.1145/3632775.3663310

1 INTRODUCTION

The ongoing shift towards sustainable energy leads to a significant restructuring of the energy sector: large-scale integration of distributed renewable energy sources, increased electrification, phasing out of fossil fuel-based generation, etc. [9]. As a result of these changes, there is a growing need for grid balancing services and demand-side flexibility to ensure the reliable and secure functioning of the grid. Conventionally, large industries and big consumers were the primary sources of such demand-side flexibility. However, another important and as-of-yet largely untapped source of energy flexibility is the residential sector [8].

Realizing a solution that unlocks this (residential) flexibility requires effective controllers that can manage the energy consumption of buildings, operating in a sequential way under uncertain operating conditions. Developing controllers for such home energy management systems (HEMS) is an extremely challenging task and has been a major research area [7, 16]. A prominent and established method in this domain is Model Predictive Control (MPC). MPC relies on a mathematical model of the system to anticipate its future behavior and an optimizer that uses this model to obtain optimal control actions [3]. Several works have demonstrated the effectiveness of MPCs in both simulation and real-world scenarios, e.g., [5, 11]. However, most MPC deployments are limited to large commercial or institutional buildings, because they strongly rely on accurate system models, which require a non-trivial effort to construct [23].

Consequently, recent research in designing controllers has shifted towards data-driven RL-based methods [2]. RL-based controllers rely on data obtained by interacting with the household, circumventing the need for bespoke models as common with MPCs. Such works show applications of RL in HEMS [13, 19], including some real-world pilot studies [25]. While these RL solutions are promising, realizing them in a commercially viable HEMS is still challenging. One of these challenges pertains to the lack of explainability associated with RL and deep RL algorithms [15, 17].

Since most HEMS are directly exposed to ordinary end-users who typically are no energy experts, the primary requirement for any acceptable control policy is that it can be easily explained to such end-users. Since usual RL-based controllers rely on deep neural networks, they are inherently black-box and hence difficult to explain [15]. Additionally, while works in explainable AI such as [18, 24] explore some post-hoc explanation techniques based on SHAP values[1] or feature importance, these explanations are mainly

[1] SHapley Additive exPlanations, [14].

aimed at machine learning experts and cannot be offered to average users (laypeople).

We identify this as a major obstacle in realizing practical, data-driven, RL-based HEMS and present our work on learning RL policies using differentiable decision tree (DDT) as a possible solution to this problem. The key idea is to replace the (deep) neural network-based control policy with a simple decision tree-based policy that is structurally explainable, i.e., in the form of rather simple *if-then-else* rules, while being able to learn such trees using data and gradient descent. Inspired by works such as [20], we demonstrate how differentiable decision trees can be used with standard state-of-the-art off-policy RL algorithms such as DDPG [12], and how such trained actors lead to explainable control policies. Concretely, the main contributions of our work are:

(1) We introduce a new 'actor' architecture based on differential decision trees to train standard off-policy actor-critic RL agents.
(2) We investigate the explainability of the DDT-based control policies for different sized trees.
(3) We demonstrate the usability of such an agent on a preliminary HEMS problem, comparing its performance against baseline and standard RL controllers.

Note that, while [20] previously introduced a similar method, their approach was restricted to Atari games and other benchmark RL domains. To the best of our knowledge, our work is (one of) the first applications of differentiable decision tree-based RL agents in the energy domain.[2]

2 METHODOLOGY

2.1 Problem Formulation

We examine our proposed DDT-based RL controller in the context of a home energy management system (HEMS), where the goal is to efficiently control a home battery (flexibility asset) to optimize the energy bill of a homeowner. As a specific case study, we consider an average Belgian household with a rooftop solar PV installation (with generated power P_t^{pv}), non-flexible electrical load (P_t^{con}), and a home battery. We assume that this household is exposed to varying BELPEX[3] day-ahead prices (λ_t^{con}) and a capacity tariff based on peak power [22]. This leads to a joint optimization problem, where the HEMS must minimize the daily cost of both the energy consumption (c_t^{eng}) and the peak power (c_t^p) (detailed in Appendix A). To realistically reflect today's typical conditions, we incorporate solar PV and consumption profiles from a real-world household and use actual BELPEX day-ahead prices from 2024.

2.2 Reinforcement Learning

We formalize the sequential decision-making problem presented in §2.1 as a Markov Decision Process (MDP) [21]. Such MDP comprises a system state representation ($\mathbf{x}_t$), a transition function f defining how the state evolves over time, based on actions (u_t), where such transitions have an associated reward/cost to be maximized/minimized. In our case, the state ($\mathbf{x}_t$) comprises the current price, battery state-of-charge, non-flexible demand (P_t^{con}), and solar PV generation (P_t^{pv}). The actions (u_t) are the charging/discharging signals for the battery. For improved explainability, we assume a discrete action space of 5 elements (i.e., $\mathbf{U} = \{-1, -0.5, 0, 0.5, 1\}$), with the possibility of extension to continuous action spaces reserved for future work. Our cost function comprises both time-varying energy cost and peak power capacity cost (c_t^{eng} and c_t^p respectively; see Appendix A). The transition function (f) models the dynamics of the household and the battery.

The goal of an RL agent is to find a policy $\pi : \mathbf{X} \rightarrow \mathbf{U}$ that minimizes the expected T-step cost starting from an initial state $\mathbf{x}_0 \in \mathbf{X}$. For our work, we focus on DDPG, a state-of-the-art off-policy, actor-critic algorithm, where the actor learns a control policy and the critic concurrently estimates the optimal state-action value function (Q-function). For more details about this algorithm, we refer to [12], with additional modifications discussed in Appendix B.

2.3 Differentiable Decision Trees

Differentiable decision trees or soft decision trees are a variant of ordinary decision trees, introduced in prior works such as [4, 10]. Like ordinary decision trees, DDTs have two types of nodes: (i) decision nodes, and (ii) leaf nodes. Decision nodes comprise feature selection weights ($\boldsymbol{\beta}$) for selecting a feature and cut-threshold (ϕ) for splitting across the selected feature. However, unlike ordinary trees, the decision node in DDTs implements a soft decision using the sigmoid function (σ) as shown in Eq. (1a). The leaf nodes contain an output distribution vector ($\mathbf{w}$) that is tuned to obtain an output probability distribution ($\mathbf{p}^L$), which in our case is the probability distribution over the (discrete, cf. supra) action space ($\mathbf{U}$). This is modeled using *softmax*, calculating the probability for each action $u_m \in \mathbf{U}$ as Eq. (1b).

$$p^{\text{left}} = \sigma(\boldsymbol{\beta}\mathbf{x} - \phi) \quad ; \quad p^{\text{right}} = 1 - \sigma(\boldsymbol{\beta}\mathbf{x} - \phi) \tag{1a}$$

$$p_m^L = \frac{e^{-w_m}}{\sum_{\kappa=1}^{|\mathbf{U}|} e^{-w_\kappa}} \qquad \forall m \in \{1, 2, \ldots, |\mathbf{U}|\} \tag{1b}$$

A DDT of arbitrary depth is then built using such decision and leaf nodes (e.g., Fig. 2). Each decision node gives the path probabilities for its edges and each leaf node gives the output probability distribution. The final tree is obtained by appropriately combining these probability values following the tree structure. As an example, §2.4 details the exact formulation of a DDT of depth 2 and its forward pass. At inference time, each decision node is converted from the 'soft' version into a 'crisp' decision by using *argmax*, *max* operators, resembling an ordinary decision tree. The trainable parameters ($\boldsymbol{\beta}$, ϕ and $\mathbf{w}$) are initialized randomly and learned via gradient descent.

2.4 Formulation of DDT of depth 2

Based on §2.3, we provide the formulation for a DDT of depth 2 (Fig. 1). The path probabilities (p_i) and leaf probabilities (p_{jk}^L) are computed using Eq. (1a) and Eq. (1b) respectively. Algorithm 1

[2] In our earlier work, we explored the application of DDTs in a policy distillation scenario. This study distinguishes itself by directly integrating the DDTs in standard actor-critic RL algorithms.

[3] This refers to the organised wholesale market for power trading in the Belgium energy market, i.e., the current European Power Exchange Belgium.

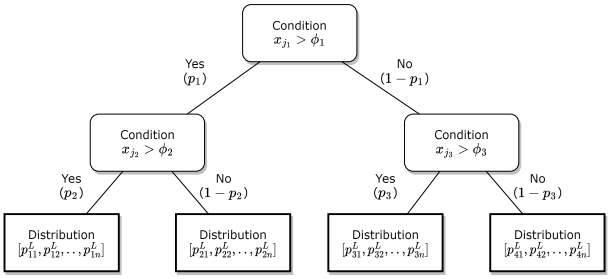

Figure 1: Illustration of a DDT of depth 2 with decision nodes denoted by rounded boxes and leaf nodes with rectangles. Here, all p_i represent path probabilities and p^L_{jk} represents probabilities at each leaf (j) for each element (k). Further, $n = |\mathbf{U}|$ represents the size of the action space.

Algorithm 1 Depth 2 DDT Formulation

1: Initialize: $\boldsymbol{\beta_i}$, $\boldsymbol{\phi}$, $\mathbf{w}^L_k$, where $i = \{1, 2, 3\}$ (decision nodes) and $k = \{1, 2, 3, 4\}$ (leaf nodes)
2: Input: State $\mathbf{x}$
3: **for all** i **do**
4: Feature Selection: $x_j = \boldsymbol{\beta_i} \cdot \mathbf{x}$
5: Evaluate Condition: $p_i = \sigma(x_j - \phi_i)$
6: **end for**
7: Calculate Path Probabilities: $\mathbf{p} = \begin{bmatrix} p_1 & 0 \\ 0 & 1-p_1 \end{bmatrix} \cdot \begin{bmatrix} p_2 & 1-p_2 \\ p_3 & 1-p_3 \end{bmatrix}$
8: **for all** k **do**
9: Calculate Leaf Probabilities: $\mathbf{p}^L_k = \{p^L_{k1}, p^L_{k2}, \dots p^L_{kn}\}$ based on Eq. (1b), where $n = |\mathbf{U}|$
10: **end for**
11: Output: $o = \mathbf{p}[1,1]\mathbf{p}^L_1 + \mathbf{p}[1,2]\mathbf{p}^L_2 + \mathbf{p}[2,1]\mathbf{p}^L_3 + \mathbf{p}[2,2]\mathbf{p}^L_4$

shows the implementation of the DDT. Thus, first the path probabilities and leaf probability distributions are computed. Subsequently, the probabilities are combined according to the tree structure: (i) For each branch (that starts from the root and ends at a leaf), path probabilities for each edge of the branch and the corresponding leaf node distributions are multiplied to get a probability distribution for that branch; (ii) Output distributions of each of the branches are added to get the final distribution as the output of the DDT. This algorithm describes the 'forward' pass of the DDT used for training. At inference, all the 'soft' operations are converted into 'crisp' operations, and the DDT is reduced to an ordinary decision tree.

3 RESULTS

We validate the performance of our proposed DDT-based RL agents on a battery-based HEMS problem (discussed in §2) and investigate the control performance and explainability of the learned controllers using 'shallow' DDTs of depth 2 and 3.

3.1 Performance of DDT-based Agents

The performance of our proposed approach using DDTs of depth 2 and 3 is presented in Table 1 (listing the mean and standard deviation over 5 seeded runs along with the minimum and maximum values). We note three key observations: (i) DDT agents of depth 3 outperform all other agents including standard DDPG; (ii) both DDT agents outperform the baseline RBC controller;[4] (iii) the performance difference between DDTs of depth 2 and standard DDPG agents is quite small (~4%). This indicates that our proposed approach can learn good control policies and outperform typical, built-in RBC included with commercially available batteries [1].

Table 1: Comparison of DDT-based agents

Algorithm	Cost		
	Mean (std deviation)	Min	Max
DDPG (Standard)	€ 3.34 ± 0.8	€ 2.29	€ 4.64
DDT (depth 2)	€ 3.47 ± 1.8	€ 1.48	€ 6.04
DDT (depth 3)	€ **3.02** ± 1.5	€ 1.48	€ 5.54
Baseline RBC	€ 4.70	–	–

Note that, although the mean performance of DDT agents of depth 3 is slightly better than the standard DDPG agents, the large standard deviation values suggest that this difference is not statistically significant. Furthermore, the large difference between the minimum and maximum values of DDT-based agents indicates some instability in the training process, with some models converging to an inferior control policy. This instability in learned models could be associated with the tree structure of the DDT, where changes in hierarchically higher decision nodes could disproportionately impact the output distributions. Future work will further investigate and address this (in)stability issue.

3.2 Explainability of DDT-based agents

Aside from control performance, we study the explainability of the learned DDT policies. The learned DDT policies can be easily visualized owing to their tree structure. As an example, Fig. 2 presents a learned policy of a DDT of depth 2. Note that these DDTs are randomly initialized and only learn the feature selection (e.g., choosing 'demand' or 'price' as the feature for the decision node) and the respective cut thresholds during training, through gradient descent. Figure 2 illustrates that the learned DDT is straightforward to understand and takes intuitive actions — e.g., the DDT policy only charges the battery when both the price and demand are low, indicating a consistent peak shaving behavior taking into account current as well as future trends. These preliminary findings clearly show that in addition to the strong performance of the DDTs, the learned policies are also easy to explain and can potentially improve user acceptance of such HEMS.

4 CONCLUSION

We introduced a novel method for obtaining explainable RL-based control policies using differentiable decision trees. The DDTs can easily 'fit' into standard actor-critic RL algorithms as shown in our implementation using DDPG on a battery-based home energy management scenario. Our results in §3 clearly demonstrate that our proposed DDT-based agents can learn high-quality control

[4] A typical, commercially installed self-consumption controller.

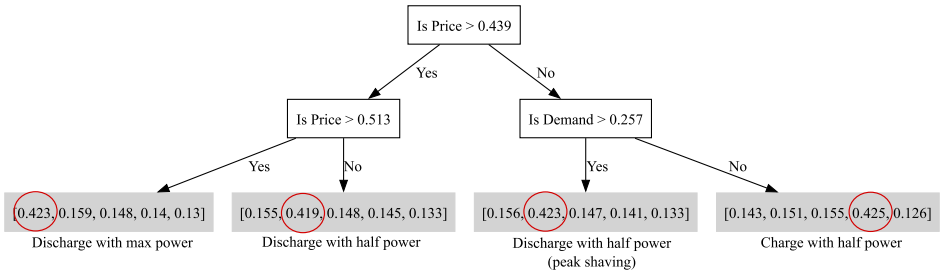

Figure 2: Example of a learned DDT for depth 2 showing selected features, learned thresholds and output distributions. The annotations indicate how the DDT can be explained.

policies while being simple and easy to explain. Preliminary findings show that the DDT-based agents lead to an overall comparable performance as compared to standard neural network-based agents and outperform them in certain settings.

4.1 Limitations and Future Work

The goal of this work was to introduce a novel method for explainable RL-based controllers for energy applications. The limitations thereof pertain to (i) the simplicity of the considered case study problem, (ii) training (in)stability, and (iii) lack of real-world validation. Future work includes extending our initial exemplary problem scenario by expanding its complexity and including different flexibility assets such as batteries, EVs and building thermal mass. The key idea will be to develop an elaborate, data-driven HEMS based on DDTs that can efficiently leverage these flexibility assets. Another fundamental research direction includes investigating the training instability of the DDT-based agents and identifying possible solutions for it (Appendix C). Besides these simulation-oriented studies, we aim to deploy such DDT-based RL in real-world households to study the performance and user acceptance of such an 'AI' driven method.

ACKNOWLEDGMENTS

This research has received funding from the Horizon 2020 Project RENergetic (grant no. 957845) and Energy Transition Fund's FlexMyHeat project.

REFERENCES

[1] AlphaESS. 2022. AlphaESS User Manual. Retrieved April 02, 2024 from https://www.alphaess.com/En/Skippower/downloadFile?id=129&mid=80
[2] Xin Chen, Guannan Qu, Yujie Tang, Steven Low, and Na Li. 2022. Reinforcement learning for selective key applications in power systems: Recent advances and future challenges. *IEEE Transactions on Smart Grid* 13, 4 (2022), 2935–2958.
[3] Ján Drgoňa, Javier Arroyo, Iago Cupeiro Figueroa, David Blum, Krzysztof Arendt, Donghun Kim, Enric Perarnau Ollé, Juraj Oravec, Michael Wetter, Draguna L Vrabie, et al. 2020. All you need to know about model predictive control for buildings. *Annual Reviews in Control* 50 (2020), 190–232.
[4] Nicholas Frosst and Geoffrey Hinton. 2017. Distilling a neural network into a soft decision tree. *arXiv preprint arXiv:1711.09784* (2017).
[5] ILR Gomes, MG Ruano, and AE Ruano. 2023. MILP-based model predictive control for home energy management systems: A real case study in Algarve, Portugal. *Energy and Buildings* 281 (2023), 112774.
[6] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. 2018. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905* (2018).
[7] Binghui Han, Younes Zahraoui, Marizan Mubin, Saad Mekhilef, Mehdi Seyedmahmoudian, and Alex Stojcevski. 2023. Home Energy Management Systems: A Review of the Concept, Architecture, and Scheduling Strategies. *IEEE Access* (2023).
[8] IEA. 2023. Energy Technology Perspectives 2023, IEA, Paris. Retrieved January 28, 2024 from https://www.iea.org/reports/energy-technology-perspectives-2023
[9] IRENA. 2023. World Energy Transitions Outlook 2023: 1.5°C Pathway, Volume 2, International Renewable Energy Agency, Abu Dhabi. Retrieved January 28, 2024 from http://ccrma.stanford.edu/~jos/bayes/bayes.html
[10] Ozan Irsoy, Olcay Taner Yıldız, and Ethem Alpaydın. 2012. Soft decision trees. In *Proceedings of the 21st international conference on pattern recognition (ICPR2012)*. IEEE, 1819–1822.
[11] Xin Jin, Kyri Baker, Dane Christensen, and Steven Isley. 2017. Foresee: A user-centric home energy management system for energy efficiency and demand response. *Applied Energy* 205 (2017), 1583–1595.
[12] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015).
[13] Paulo Lissa, Conor Deane, Michael Schukat, Federico Seri, Marcus Keane, and Enda Barrett. 2021. Deep reinforcement learning for home energy management system control. *Energy and AI* 3 (2021), 100043.
[14] Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Proc. Advances in Neural Information Processing Systems (NIPS 2027)*, Vol. 30. Long Beach, CA, USA, 1–10.
[15] R Machlev, L Heistrene, M Perl, KY Levy, J Belikov, S Mannor, and Y Levron. 2022. Explainable Artificial Intelligence (XAI) techniques for energy and power systems: Review, challenges and opportunities. *Energy and AI* 9 (2022), 100169.
[16] Bandana Mahapatra and Anand Nayyar. 2022. Home energy management system (HEMS): Concept, architecture, infrastructure, challenges and energy management schemes. *Energy Systems* 13, 3 (2022), 643–669.
[17] Zoltan Nagy, Gregor Henze, Sourav Dey, Javier Arroyo, Lieve Helsen, Xiangyu Zhang, Bingqing Chen, Kadir Amasyali, Kuldeep Kurte, Ahmed Zamzam, et al. 2023. Ten questions concerning reinforcement learning for building energy management. *Building and Environment* (2023), 110435.
[18] Meike Nauta, Jan Trienes, Shreyasi Pathak, Elisa Nguyen, Michelle Peters, Yasmin Schmitt, Jörg Schlötterer, Maurice van Keulen, and Christin Seifert. 2023. From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable ai. *Comput. Surveys* 55, 13s (2023), 1–42.
[19] Thijs Peirelinck, Chris Hermans, Fred Spiessens, and Geert Deconinck. 2024. Combined peak reduction and self-consumption using proximal policy optimisation. *Energy and AI* 16 (2024), 100323.
[20] Andrew Silva, Matthew Gombolay, Taylor Killian, Ivan Jimenez, and Sung-Hyun Son. 2020. Optimization methods for interpretable differentiable decision trees applied to reinforcement learning. In *International conference on artificial intelligence and statistics*. PMLR, 1855–1865.
[21] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.
[22] VREG. 2023. What is the capacity rate and how is it calculated? Retrieved January 29, 2024 from https://www.vreg.be/nl/wat-zijn-de-nieuwe-nettarieven-en-hoe-worden-ze-berekend
[23] Eva Žáčeková, Zdeněk Váňa, and Jiří Cigler. 2014. Towards the real-life implementation of MPC for an office building: Identification issues. *Applied Energy* 135 (2014), 53–62.
[24] Ke Zhang, Jun Zhang, Pei-Dong Xu, Tianlu Gao, and David Wenzhong Gao. 2021. Explainable AI in deep reinforcement learning models for power system emergency control. *IEEE Transactions on Computational Social Systems* 9, 2 (2021), 419–427.
[25] Xiangyu Zhang, Xin Jin, Charles Tripp, David J Biagioni, Peter Graf, and Huaiguang Jiang. 2020. Transferable reinforcement learning for smart homes. In *Proceedings of the 1st international workshop on reinforcement learning for energy management in buildings & cities*. 43–47.

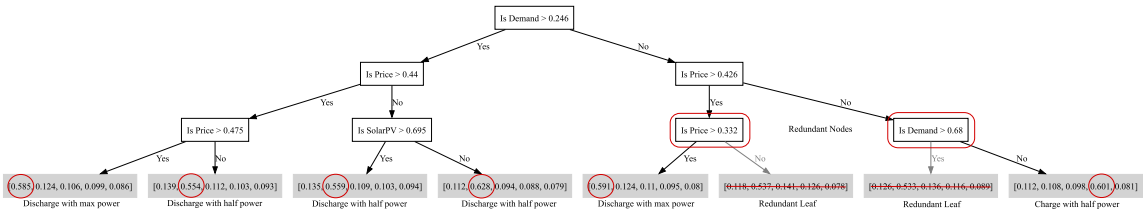

Figure 3: Example of a learned DDT for depth 3

Table 2: Parameters related to the Battery model used in the Home Energy Management Simulator

Parameter	Value
Max Capacity	10 kWh
Max Power	4 kW
Efficiency (round trip)	0.9
Action Space	$\{-1, -0.5, 0, 0.5, 1\}$

APPENDIX

A HOME ENERGY MANAGEMENT PROBLEM

The home energy management problem described in §2.1 is modeled as:

$$\min_{u_1,\dots u_T} \sum_{t=1}^{T} c_t^{\text{eng}} + c_t^{p} \tag{2a}$$

$$\text{s.t.: } c_t^{\text{eng}} = \begin{cases} \lambda_t^{\text{con}}\, P_t^{\text{agg}}\, \Delta t & : P_t^{\text{agg}} \geq 0 \\ \lambda_t^{\text{inj}}\, P_t^{\text{agg}}\, \Delta t & : P_t^{\text{agg}} < 0 \end{cases} \quad \forall t \tag{2b}$$

$$c_t^{p} = \lambda^{\text{cap}}\ \max(P_t^{\text{agg}},\ P_{\min}^{agg}) \tag{2c}$$

$$P_t^{\text{agg}} = P_t^{\text{con}} + P_t^{\text{pv}} + u_t \quad \forall t \tag{2d}$$

$$E_{t+1} = \begin{cases} E_t + \eta\, u_t\, \Delta t & : u_t \geq 0 \\ E_t + \frac{1}{\eta}\, u_t\, \Delta t & : u_t < 0 \end{cases} \quad \forall t \tag{2e}$$

$$0 \leq E_t \leq E^{\max};\ u^{\min} \leq u_t \leq u^{\max} \quad \forall t. \tag{2f}$$

The battery is modeled using a linear model (Eq. (2e) with charging/discharging actions u_t and current energy level (E_t). The cost of energy consumed (c_t^{eng}) depends on the actual power consumed (P_t^{agg}) and the current injection and consumption prices (λ_t^{inj} and λ_t^{con} respectively). Similarly, the capacity cost (c_t^{p}) depends on the actual power consumed and the minimum power capacity contracted (which is set to 4kW). Furthermore, we assume $T = 24$ hours and a time resolution $\Delta t = 1$ hour. The battery configuration is listed in Table 2. The RBC tries to maximize self-consumption as common with commercially available batteries.

B DISCRETE ACTOR-BASED DDPG

As described in §2, we use DDTs as the actor-network in DDPG. However, standard DDPG implementations work only with continuous actions. To overcome this challenge, we implement a discrete actor-based DDPG agent, inspired by [6]. The key changes are: (i) the target values for critic training are computed using Eq. (3); (ii) the equation for computing actor loss (as gradient of Q-values) is modified to Eq. (4).

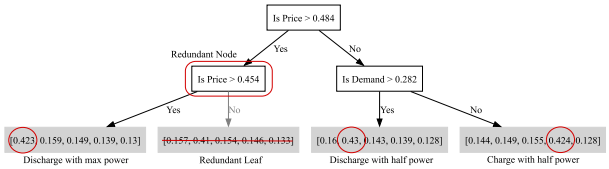

Figure 4: Example of a learned DDT for depth 2 with a redundant decision node

$$\text{critic target} = \left(c_i + \sum_{k=1}^{|U|} p(u_k|\mathbf{x}_{i+1}) \hat{Q}_{\theta_c^-}(\mathbf{x}_{i+1}, u_k) \right) \tag{3}$$

$$\mathcal{L}_a = \nabla\, \mathbb{E}\left[\sum_{k=1}^{|U|} p(u_k|\mathbf{x}_i) \hat{Q}_{\theta_c}(\mathbf{x}_i, u_k) \right] \tag{4}$$

C ADDITIONAL RESULTS

C.1 Depth 2 DDT with redundant rules

In some instances, the decision nodes in the DDTs learn redundant or conflicting rules, which in some cases can lead to inferior results. An example of such a DDT is depicted in Fig. 4. Here, the highlighted decision node cuts along the same feature as its parent node and learns a redundant threshold leading to one of the leaves being unreachable. This can contribute to training instability and needs to be investigated further.

C.2 Depth 3 DDT

Besides DDTs of depth 2, we also trained DDTs with depth 3. An example of such a decision tree is shown in Fig. 3. The higher depth enables this variant to have more decision nodes, leading to a more complex tree representation that can better capture the environment dynamics. However, as observed in Fig. 3, not all leaf nodes are being used due to the decision nodes learning redundant rules, similar to the tree depicted in Fig. 4. This indicates the need for further tuning the training process and/or introducing additional loss terms that can penalize such behavior.