Eberhard Karls Universität Tübingen
Mathematisch-Naturwissenschaftliche Fakultät
Wilhelm-Schickard-Institut für Informatik

Master Thesis Computer Science

Uncertainty-Aware Reinforcement Learning for Demand Response in Energy Systems

Ludwig Bald

31. Januar 2023

Reviewers

Dr. Nicole Ludwig
Wilhelm-Schickard-Institut für Informatik
Universität Tübingen

Jun. Prof. Dr.-Ing.
Setareh Maghsudi
Wilhelm-Schickard-Institut für Informatik
Universität Tübingen

Bald, Ludwig:
Uncertainty-Aware Reinforcement Learning for Demand Response in Energy Systems
Master Thesis Computer Science
Eberhard Karls Universität Tübingen
Thesis period: July 2022-January 2023

Abstract

Write here your abstract.

Zusammenfassung

Bei einer englischen Masterarbeit muss zusätzlich eine deutsche Zusammenfassung verfasst werden.

Acknowledgements

Write here your acknowledgements.

Contents

Chapter 1

Introduction

Start with a comprehensive introduction about the questions of your thesis. 1-2 pages:

When proofreading: Check that all terms and abbreviations are introduced here

Climate Change is the global challenge of our lifetime. Carbon introduced into the atmosphere when burning fossil fuels for human needs causes global warming, destabilizing the climate, ecosystems, and societies around the world. In the Paris Agreement of ... governments have committed to an ambitious goal of drastically reducing carbon emissions to keep global warming from increasing beyond 2°C, compared to ... The latest IPCCC report urges governments to take stronger actions, or their previous commitment will not be reached. A key strategy for reducing carbon emissions from a range of sources is the combination of two measures: The first step is to electrify current processes that use fossil fuels, like replacing gas-fired furnaces with heat pumps. The second step is to replace carbon-intensive electricity generation with renewable options like solar and wind power.

cite

cite

While much better for the natural environment, renewable sources of energy pose a challenge for a grid built for fossil fuels: Unlike fossil-fuelled power plants, renewable power production depends on the weather, and it can not react flexibly to changes in demand.

As the share of installed renewable sources of electricity continues to grow from today's...%, the reliability of the electricity supply goes down.

cite

talk about reacting to less reliability in renewable production, not only reacting to demand

In order to keep the grid stable, supply and demand must always be in balance. Before the green transition, this was achieved by flexible power generation: When demand was high, electricity producers were able to react and increase production. This was incentivized by a complex and tightly regulated market constructed on top of the physical layer. As the share of renewable power increases, fossil-fuelled plants remain the only market participants that can flexibly react to changes in demand. When phasing out fossil-fuelled power generation, this flexibility needs to be provided by different parts of the system.

There are dedicated electricity storage facilities that can react very quickly to stabilize the grid by storing and releasing energy as needed. However, consumer electricity demand can flexibly react to changes in supply, a scheme called Demand Response. Grid-scale storage in Europe mainly consists of hydropower, which has been installed where mountainous geography allows, and capacity has reached its natural limit. More expensive battery-powered storage facilities are slowly being built, but are largely not cost-efficient in the current economic setting.

cite

In this thesis, I focus on an opportunity to make consumer electricity demand more flexible.

Buildings require energy mainly for heating and cooling the air and the water supply. Today, they are responsible for ...% of total energy demand. On the other hand, buildings often contribute to electricity production through photovoltaic panels. New buildings often come with a battery, which enables them to more efficiently use their solar electricity.

cite

Building's electricity consumption is already largely automated and is therefore a prime candidate for automated demand response.

mention incentive-based human DR

The Reinforcement Learning family of control algorithms has successfully been applied for control of the battery of simulated buildings. . A popular simulation framework for this purpose is CityLearn. In this thesis, I set out to test Uncertainty-Aware Deep Q-Networks on CityLearn. UA-DQN is an uncertainty-aware adaptation of Deep Q-Learning.

list other automated control algorithms and their goals

cite algorithms

cite CityLearn

cite algorithm

cite DQN

The algorithm's better treatment of uncertainty should lead to overall better performance with less need for data, better robustness for novel data, and can even be leveraged for differently risk-aware charging and discharging strategies.

talk about results!

These aspects are important for a demand response algorithm, which provides flexibility to the grid while reliably meeting building demands for energy.

rephrase

This thesis is structured as follows: In the following chapter, I motivate in more detail the need for Automated Demand Response. I introduce the theory of Reinforcement Learning and lay the foundations for the uncertainty-aware algorithm. In chapter 3, I present the uncertainty-aware algorithm, as well as a detailed description of the experimental setup. I present the results in chapter 4. A discussion and a short outlook conclude the thesis.

Things to add in introduction: - Einordnung in die bestehende Literatur. Was ist an meinem Ansatz anders als an bisherigen Ansätzen? - Little bit of Results (abstrakt success vs failure) (technical, numbers)

Terms to check: - Demand Response vs. Demand Side Management - Reinforcement Learning - Electrical Grid

Chapter 2

Background

Give all the necessary background that is needed to justify and understand the problem and the approach. - comment on employed hardware and software - describe methods and techniques that build the basis of your work - review related work(!) - roughly 1/3 of thesis

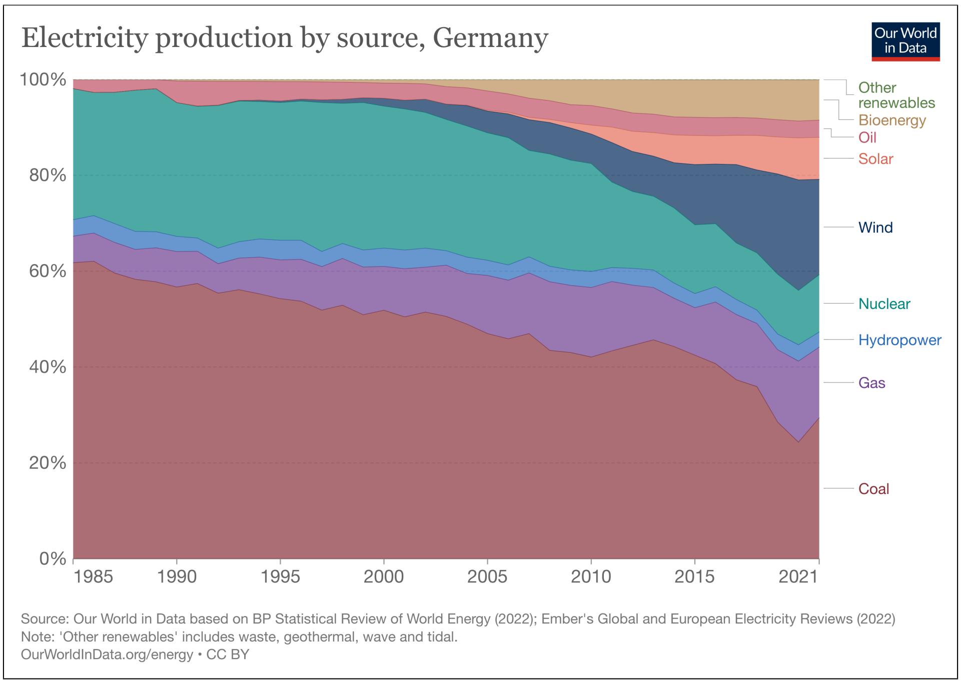
2.1 The Electrical Grid and Flexibility

The electrical grid is basic infrastructure that enables the function of our modern society. It connects electricity consumers, from private consumers to heavy industry, with producers. Historically, the entire system has been built for reliable large-scale power generation, overwhelmingly fuelled by coal and natural gas, together accounting for 67% of Germany's electricity consumption in 1985, with 27% provided by nuclear energy Ritchie et al. (2022).

Climate change and the exit from nuclear power require a radical increase in the share of renewable electricity. As of 2021, renewable energy accounts for 40% of electricity consumption in Germany Ritchie et al. (2022). An overview of the historical development is shown in figure 2.1

While conventional energy generation can flexibly respond to demand, renewable energy depends on the weather. It is therefore intermittent and harder to predict, see figure 2.2. To be able to meet demand, there is a new need for flexible backup power. Natural gas plants are more environmentally friendly than coal plants and can be flexibly turned on and off in a matter of minutes. As a result, the share of electricity powered by natural gas has risen along with renewables.

Another aspect of the energy transition is the electrification of many processes which previously used fossil fuels directly. Internal combustion engine cars are being replaced by more efficient battery-powered electric cars and heating units that use natural gas or oil are being replaced by much more



Electricity production by source, Germany

Source: Our World in Data based on BP Statistical Review of World Energy (2022); Ember's Global and European Electricity Reviews (2022)
Note: 'Other renewables' includes waste, geothermal, wave and tidal.
OurWorldInData.org/energy • CC BY

Figure 2.1: The share of renewable electricity has risen significantly from 1985 to 2021, while nuclear and fossil fuels have declined.

efficient electric heat pumps. Many heat pumps can also be run in reverse to cool a building, which climate change makes more and more necessary, but this enables additional electricity demand in the summer.

The ongoing energy transition puts a lot of pressure on the grid. On the one hand, the electricity supply is becoming less reactive, while on the other hand, many polluting processes are being electrified, causing additional demand. During large future demand spikes, the total load on the grid could exceed current capacity. To cope, new transmission lines are being built, a costly and complicated process.

maybe give specific number

In order to both reduce reliance on fossil fuels and to keep the total load below grid capacity, there is a need for additional flexibility in the system. A traditional source of flexibility has been pumped hydropower storage. When there's unused electricity, it can be stored by pumping water uphill. When needed, water can be released and used to generate electricity. Hydropower depends almost entirely on geography. There is almost no potential to develop more hydropower storage.

A different large-scale method of energy storage is batteries, which have only found limited use due to their cost. Hydrogen can also be used as a way to store and even transport energy, with several chemical processes currently being explored. However, this method is also costly, and any produced hydrogen is probably worth more as a crucial chemical than as pure electricity

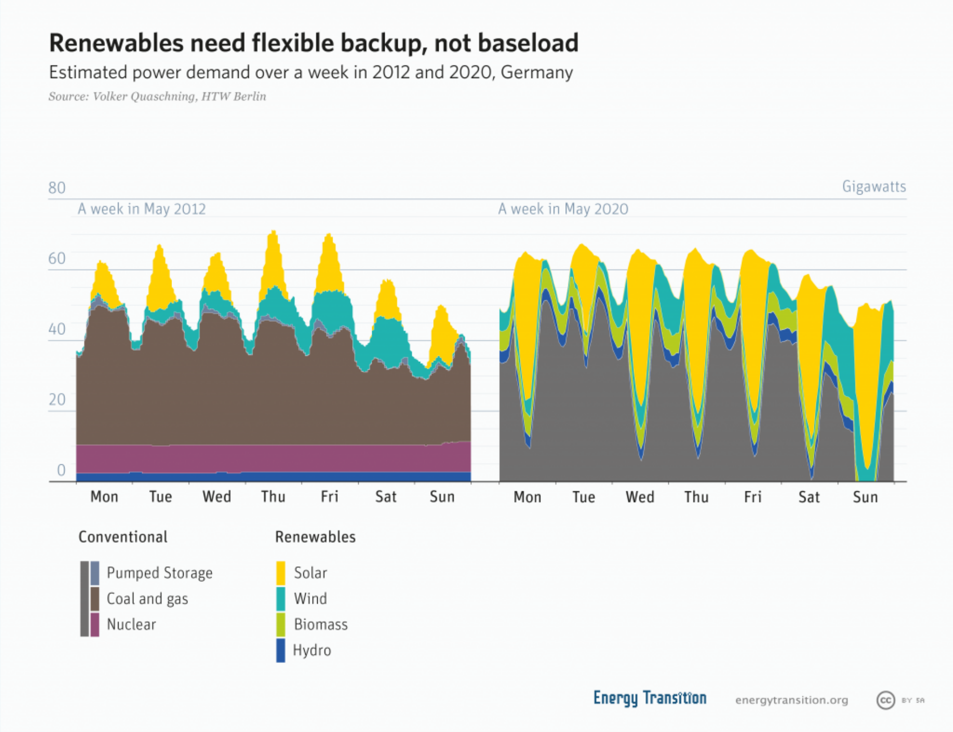


Figure 2.2: Renewable electricity is intermittent and hard to predict precisely. As the share of renewable electricity increases, other sources of electricity need to respond flexibly to meet demand.

storage.

rephrase, cite

As flexibility in electricity supply decreases, and centralized storage facilities remain too expensive, electricity demand needs to become more flexible.

2.2 Demand Response

Both the inflexibility of renewable power generation and the limited capacity for centralized flexible energy storage leave one component of the electric system: In a renewable-dominated grid, demand needs to flexibly respond to changes in available supply. In order to stabilize the grid, grid operators employ schemes to curb demand in case of exceptionally large pressure on the grid. Large industrial consumers are paid in advance for shutting off their processes if needed. As a matter of last resort, rolling blackouts are introduced to curb demand when electricity production can not keep up with consumption. An electric grid designed for renewable energy needs more fine-grained coordination across a larger fraction of demand.

In order to implement demand response, electricity consumers need to be incentivized and able to adapt their processes to available supply. Proposed incentive schemes for demand response include market-based solutions that feature a flexible electricity price and solutions of centralized control that pay out flat rewards for participation, as well as combinations of the two. In order to be able to react to changes in demand, electricity-consuming processes need to be aware of the current and future available supply. This can be a human in the loop, deciding to shift a process to a time with cheaper electricity, or this can be automated. For example, a private prosumer that produces solar electricity might prefer to run their washing machine only on sunny days when electricity is free to them. However, this means the human needs to keep track of the weather and is put under additional cognitive load when planning this.

cite

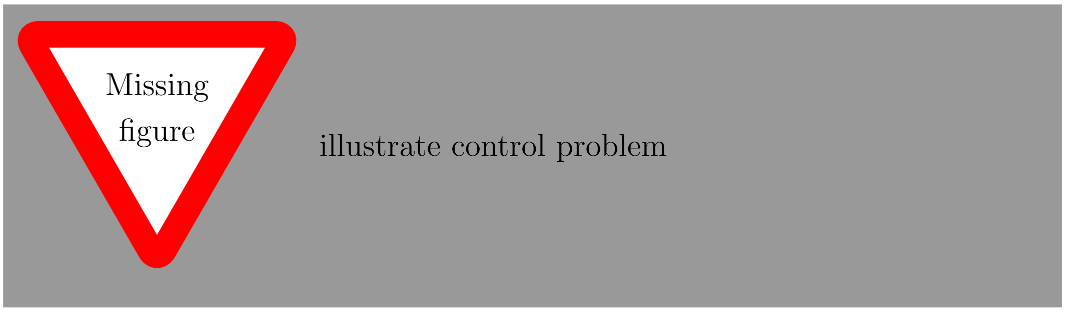
As stated before, the process of heating is being electrified. While this increases the load on the electric grid, it is also a chance to provide flexibility: When equipped with an intelligent and connected control system, the heating system can flexibly react to changes in price, using hot water tanks or the building itself as heat storage. Similarly, when intelligently automated, the charging process of an electric car or a home battery can somewhat react to market conditions.

somehow mention coordination goals

Technologically, this amounts to a control problem: The controller's goals are to meet certain demands (like ensuring a comfortable living temperature) while minimizing cost. In the context of the larger system, there is the shared goal of coordination between different electricity-consuming processes. This involves both an understanding of the dynamics of the controlled system and an understanding of how prices are likely to change. Different technological approaches can be employed for this. When system dynamics are known and prices change predictably, for example with fixed rates per time of day, an optimal rule-based controller can be derived. A rule-based controller has the advantage of being transparent and reliable, but it's not flexible enough to react to changing system and pricing dynamics. Several adaptive control algorithms have been proposed for use in demand response.

cite and make sure this is true!

List some and cite



notes on this section: - focus more on buildings - use the following review paper: - Li et al. (2021), a review paper about energy flexibility in residential buildings

2.3 Reinforcement Learning

1. Explain RL Fundamentals 1. Markov Decision Process 1. Highlight stochasticity of transition and Reward functions 2. Reward function? 3. Mention generalizations: POMDP, (Multi-Agent MDP?), Non-stationary MDP 2. Value Learning - Introduce the notion of a state value - Bellman updates - Q-Learning - Mention Policy Gradient methods 3. Deep Q-Learning - introduce Fundamentals of Deep Learning? Deep RL is somewhat like using a POMDP? - Introduce further tricks employed by DQN - Replay Buffer - Dueling Q-Networks? - TD-Learning / Target Network 4. Importance of Action Selection and Exploration strategies - Exploration vs. Exploitation Dilemma - Convergence guarantees

2.3.1 Reinforcement Learning Fundamentals

Reinforcement Learning (RL) is a feedback-based learning paradigm derived from behavior learning in animals. This section serves as a brief introduction to the topic. Unless otherwise stated, this section is based on the textbook Sutton and Barto (2018). In Reinforcement Learning, there is a clear distinction between the learning agent and the environment. The agent is able to observe the state of the environment and perform an action. In turn, the environment is affected by the action and transitions into a new state according to its stochastic transition dynamics. The environment passes the resulting state and a reward signal back to the agent. Typically, the agent's goal is to select actions that obtain the maximum reward. In an infinite environment, the objective is to maximize the expected value of the total discounted future reward.

The environment specifies the entire reinforcement learning task. Formally, it is a discounted Markov Decision Process (MDP):

$$\text{MDP} = (S, A, R, \gamma, p),$$

where S is the set of possible states, A is the set of possible actions, R is the set of possible rewards, γ is the discount rate and $p(s', r|s, a)$ is the probability distribution that specifies the environment dynamics. It is important to note that an MDP has the Markov Property, i.e. the dynamics depend entirely on the state and action, there is no hidden state. The state space can therefore also be called the observation space.

When modeling a real-world control problem, an MDP necessarily is a simplifying assumption. In reality, state transitions often depend on outside influences or are non-stationary for other reasons. In complex problems, the desired behavior is not obvious. Therefore, designing the reward function is often non-trivial.

Mention a few RL applications that can be used as examples

Some environment states are preferable to others. Value Learning is a class of RL algorithms that builds on this intuition. From a trajectory of state-action-observation-reward tuples that were generated while following a policy π, a Value Learning algorithm assigns each state the expected future discounted reward that will be received when the agent, in state s, continues to act according to π:

$$v_\pi(s) = \mathbb{E}[G_t|S_t = s] = \mathbb{E}[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s], \text{for all } s \in S$$

Similarly, one can define the value of taking an action in a certain state:

$$q_\pi(s, a) = \mathbb{E}[G_t|S_t = s, A_t = a] = \mathbb{E}[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s, A_t = a]$$

This Action-Value Function or Q-function induces a greedy policy by evaluating $q(s, a)$ for the current state and all possible actions, and selecting the highest-expected reward action. For the optimal policy π_*, the greedy policy induced by q_{π_*} is π_* itself. For more detail on how the optimal policy can be approximated, please refer to Sutton and Barto (2018).

To iteratively improve an initial Q-function, the greedy policy induced by it can be used to generate new trajectories to learn from. It intuitively serves as an approximation to π_*. However, a slight modification needs to be made: To make sure all states are visited, the ϵ-greedy policy is used instead, which acts randomly with a small probability of ϵ, and acts greedily otherwise.

2.3.2 Deep Q-Networks

Mnih et al. (2015) propose the Deep Q-Network algorithm (DQN), which is able to learn and play many video games better than humans. DQN is a Q-Learning algorithm. It approximates the Q-function using a deep convolutional neural network, which is able to store complex information like video game dynamics.

In order to arrive at an efficient algorithm that shows stable performance in practice, they employ additional tricks. Firstly, a replay buffer stores past trajectories

- Experience Replay - Target Q-Network

This progress stems mainly from the ability of neural networks to accurately model highly complex functions like environment dynamics.

2. Related Work 1. Uncertainty-Aware Reinforcement Learning: Related Work - Uncertainty-Aware Machine Learning Methods: Distributions instead of estimators - Explicitly model input uncertainty. - Assume a distribution, model parameters - e.g. Discrete distribution, normal distribution, beta distribution, or general distribution parameterized by quantiles. - Example in RL: POMDP - Algorithms: Bayesian NN, Gaussian Processes, Variational Autoencoders (uncertainty about deep representation) - Advantages: More robust - Disadvantages: More expensive, need to model data generation process - Explicitly model output uncertainty. - In classification: Confidence score (discrete distribution), class probabilities - In regression: Again, model distribution and parameters - Examples: Bayesian NN, Ensemble models, Distribution Learning, *UA-DQN* - Advantages: More trustworthy and more useful for risk-aware decision-making, more robust, can be more sample-efficient - Disadvantages: more expensive, more complex, less robust - Examples of Uncertainty-aware RL applications: - TODO 2. Reinforcement Learning for Consumer Building Demand Response 1. Mention Frameworks, available data, including CityLearn, real-life applications 2. Mention related work: applied algorithms and their results -

——- Erster Entwurf below ——

2.3.3 Flexibility (/Demand Response)

This section introduces the notion of flexibility and therefore motivates demand response. It should also maybe talk about conflicts of interest, and incentives.
need to be defined before this section: - conventional power - renewable power - supply/demand vs production/use

- flexibility is needed both in the market layer and in the physical layer

Renewable electricity is less flexible than conventional electricity production, therefore there is a need for more flexibility elsewhere in the system. Electricity production and electricity use must always match. If there is more or less electricity used than is available, the entire grid is no longer stable. Partly, this coordination problem between producers and consumers is solved by the electricity market: An electricity producer will produce and feed to the grid only as much electricity as they can sell. This system works fine if the electricity producer can freely regulate how much power they produce based on the market.

This is not true for renewable energy, where production capacity is both determined by external factors and harder to predict. Renewable sources of electricity are therefore less flexible. In order to keep production and consumption in balance, there needs to be a system component with enough flexibility to respond to sudden changes in electricity supply and demand. This can be a hydroelectric or grid-scale battery storage system which can both produce and consume electricity. It can also be a conventional power plant that's kept in spinning reserve, to take over when there's less renewable electricity than needed. This causes carbon emissions. Often, renewable power plants can't feed all of their production into the grid, because they can't find a flexible buyer for unexpected production. This motivates the need for flexibility on the demand side.

define

2.3.4 Demand Response

- a scheme to operate the grid more efficiently as demand rises quickly. - breaks with one of the central guarantees: electricity is freely available, accepting that renewable energy is much less reliable. - Mechanism: Incentivize consumers to use electricity when there's capacity. - implementations: TODO examples (Large-scale AC cuts, incentive programs, etc.) - already in place for large industrial consumers who buy electricity on the markets. (TODO: Is that true?) - some industrial consumers are paid for providing flexibility, they can stop their processes if there's not enough electricity being produced. - more difficult for private consumers, who don't want to think about efficiency all the time. Solution: Automate where possible.

2.4 Reinforcement Learning

2.4.1 Fundamentals: Markov Decision Process

Many animals are able to learn complex behaviors by performing an action, observing the results and - if the action got them closer to their goal - repeating the action when facing a similar situation. For example, consider a food-dispensing lever in a hamster's cage. At first, the hamster might not notice the lever. But sooner or later, by accident or out of curiosity, the hamster will push the lever, dispensing an item of food. After a few repetitions, the hamster will know to walk towards the lever, and push the lever when it wants food. Our hamster uses a biological implementation of *Reinforcement Learning (RL)*. In an RL environment, an *agent* is able to repeatedly perform an *action*, observe the consequences and be rewarded or punished. RL therefore is an adaptive approach to solve feedback-based control problems.

Formally, RL environments are *Markov Decision Processes (MDP)*:

$$\text{MDP} = (S, A, p)$$

where S is the set of possible states the environment can be in, A is the set of actions available to the agent, and the transition distribution $p(s', r \mid s, a)$ specifies the environment dynamics, giving the joint probability of transitioning from state $s \in S$ to state $s' \in S$ after performing action $a \in A$, and getting reward $r \in \mathbb{R}$.

Interacting with the MDP, an agent first observes the environment's state s, then takes an action a, and then the next state s' of the environment is sampled from p. There are no further restrictions placed on the structure of state and action spaces. They can be merely labelled or ordered, finite or infinite, single- or multidimensional. Usually the agent observes a collection of variables and chooses an action along one or several dimensions.

Crucially, the states have the *Markov Property*: The probability of transitioning to state s' only depends on the current state s and action a. The history of past states does not matter, and there are no hidden facts that could change the probabilities. In practice, this is a simplifying modelling assumption. Returning to our hamster: In reality, the food-dispensing mechanism runs out of food after being activated a number of times. However, modelling the situation as an MDP, we do not keep track of history, so we can't tell in advance whether pushing the lever will actually produce food. Instead, we assume there's a certain probability of the action being unsuccessful, that is $p(s', \text{food} \mid s, \text{lever pushed}) < 1$. One generalization of the MDP that does enable us to model such hidden variables is the Partially Observable MDP, which I will cover in more detail later on .

cite section

2.4.2 Reinforcement Learning

introduce RL terms and algorithms

Let's now turn to the question how to solve a Markov Decision Process. Solving an MDP usually means finding the sequence of actions that maximize the expected total reward over all future time steps.

todo:

- episode - policy - value function - exploration vs exploitation - Bellmann Update

Baseline algorithm: - One RL algorithm, which I use as a baseline, is ??? - Explain algorithm

which one? depends on my treatment of uncertainty in the other one, so on the approach

2.5 Uncertainty in Reinforcement Learning

Introduce Uncertainty terms and formalisms from different perspectives. Then apply to RL.

There is a rich body of work on uncertainty. Mathematical and statistical notions of uncertainty, perspectives from economics for decision making under uncertainty.

Notes:

- Motivation: Most information is uncertain to some extent. Making good decisions under uncertainty requires an awareness of the uncertainty. - Uncertainty vs Risk: Uncertainty is a measure of the information content of a random variable or an observation???, Risk is the cost associated with different situations. - formal framework (maybe borrow from Econ: When to buy or sell a given asset?) - Decision making under uncertainty - which objective (Expected value vs risk metrics) - different types of uncertainty (e.g. aleatoric vs epistemic) - there are different types of uncertainty: Some uncertainty can be reduced by learning more about the problem, other uncertainty can not. - this stems from the formulation of RL as a stochastic MDP - for example, a biased coin. You will be able to learn something about it, but not actually predict the outcome ???

Uncertainty in RL: (maybe this should already be in approach?) - How is Uncertainty commonly modelled? - epistemic uncertainty in the observations: not explicitly modelled, somewhat represented in state-value function

- stochasticity in the environment dynamics (+consequences of actions): accepted in the MDP. learned as transition probabilities in model-based RL, subsumed in e.g. Q-function in model-free RL. - epistemic uncertainty in the environment dynamics: modelled as transition probabilities - stochasticity in the reward function: not usually explicitly modelled - epistemic uncertainty in the reward function: modelled implicitly in the state-value function - uncertainty about causality? - probably not really relevant? should I discuss it somewhere else? - Adaptations for explicit treatment of uncertainty: - Formalism for non-perfect observations: POMDP (usually there are hidden variables) -¿ Usually solved by estimating an MDP, solving that. - POMDP does not assume the Markov property on observations, but does assume a hidden MDP - ??? - RL from human preferences? (for learning a reward function) - Benefits of explicit treatment of uncertainty/Motivation: - Risk-aware strategies - better performance (maybe? TODO: Test this) - more robustness (possibly? TODO: support this or not) - better interpretability (possibly?) - TODO: other - Drawbacks: - more complex models require more training data - less efficient algorithms - not as well understood theoretically - might perform worse than just learning everything implicitly!

Risk-aware strategies: - can either specify a risk tolerance at time of inference or during training - during training: change reward function - at time of inference: requires model of the environment (I think) or a Q-function - more robust: can hand over control to e.g. humans when uncertain

Uncertainty in Multi-Agent Learning: (maybe exclude this completely) - Multi-Agent Environments are characterized by simultaneous actions by multiple agents, who each learn and act according to their own rewards. - More realistic and resilient than centralized control - absent trust, might be stuck in a suboptimal equilibrium

Chapter 3

Methods and Approach

Roughly 1/3 of thesis

0. Short summary of the whole experiment. - restate the research question, and how I set out to answer it. - Mention goal, algorithms, methodology. - Make the connection from work mentioned in background chapter.

- !!! where does uncertainty come from in CityLearn !!!, - forecasting problems: - uncertainty about future occupant behavior (electricity demand) - uncertainty about future solar power production (weather forecasts) - uncertainty about future costs (price forecasts) - measurement uncertainty: - observations might be imprecise - coordination uncertainty: - uncertainty about other actors' strategy and current actions - reward uncertainty: - uncertainty about the consequences of actions - the observed reward might be imprecise - the observed reward might not be the actual desired reward ???
Which uncertainties need to be specially treated?

What advantages would an uncertainty-aware strategy have?

- better risk management (possibly) - better performance (possibly) - better interpretability and robustness (possibly)

3.1 Uncertainty-Aware Deep Q-Network

1. Introduce and explain UA-DQN - High-level idea, and significance - Very good performance in original paper - With energy, you need to make risk-aware decisions - Explain idea, compared to DQN - Start with Learning the Distribution instead of Q-Values - There are two uncertainties: aleatoric and epistemic. - explain their significance - How to estimate uncertainties - How does action selection work? - Explain algorithm parameters: - ”Risk aversion” exposes the exploration vs exploitation dilemma directly - Implementation Details: - Implementation taken from the original paper - Mention Network size - Weight Scale changed - Initialization fixed - Mention all tricks used by the algorithm - replay buffer etc.

3.2 Environment

cite

The Reinforcement Learning environment used by the experiments in this thesis is based on the 2022 CityLearn Challenge. The environment provides a simulation of building energy systems, along with hourly data on electricity price, usage, solar production and weather data. The task is to control the storage and release of electricity in an electrical battery, with the objective of jointly reducing total per-building cost and carbon emissions. In contrast to the complete challenge setup, the experiments described in this thesis only use one building.

The simulation framework used by both the challenge and this thesis is CityLearn vaz (2019).

3.2.1 Data

The Dataset provided for the 2022 CityLearn challenge setup contains a year of hourly observations of a number of variables that describe the energy system of a building. At it's core, it supplies real-world per-building measurements of electricity use and photovoltaic solar power generation from five model buildings set up by the Electric Power Research Institute (EPRI) in Fontana, California as part of the research described in Narayanamurthy et al.. These are coupled with weather variables (Outdoor Temperature, Relative Humidity, Diffuse and Direct Solar Radiation). Future weather data is also provided as part of the data, with an offset of 6, 12 and 24 hours into the future. The data also contains carbon intensity and price of electricity provided by the grid. Time variables included are hour of the day, day of the week, month and whether there is a holiday. The source for the non-building data is unfortunately not given by the challenge organizers.

- Description: How does the data look? Show graphs, or reference graphs in appendix.
- Maybe: show a sample week of building data - Maybe: show a sample week of weather and CO2 data - Maybe: show seasonal variations

3.2.2 Environment Details

Observation Space

For this research, I make available a subset of the provided dimensions, given in table 3.1. Observations are dynamically normalized to zero mean and unit variance.

Table 3.1: The observation space available in the experiments

Variable	Unit
Hour	1-hot encoding 0-24
Direct Solar Irradiance (predicted 6h)	W/m^2
Carbon Intensity	kg/kWh
Building Electric Load	kW
Building Solar Generation	kW
Building Battery State of Charge	kWh

Action Space

The action space provided by CityLearn is the continuous real interval $[-1, 1]$, where negative actions are an attempt to discharge, and positive actions are an attempt to charge the battery. The action is scaled in units of the battery's capacity, so -1 means an attempt to discharge the whole battery.

The actual resulting charging and discharging speeds are limited by CityLearn's energy model and depend on the battery's state of charge.

The studied Reinforcement Learning algorithms require a discrete action space. I discretize the action space into a number of discrete actions. The number of discrete actions is determined with the experiment described in section 3.3.2.

Reward Function

The reward function is designed to match the initial challenge objective as closely as possible. The per-building reward at time step t is given by

$$r_t = -\frac{\text{cost}_t}{\text{cost}_{\text{no battery total}}} - \frac{\text{carbon emissions}_t}{\text{carbon emissions}_{\text{no battery total}}} \cdot 8760,$$

where $\text{cost}_{\text{no battery total}}$ and $\text{carbon emissions}_{\text{no battery total}}$ are the total dollar cost and carbon emissions observed over one year in a control situation where the battery is never used. The reward is always negative.

3.2.3 Implementation Details

During the 2022 CityLearn Challenge, I contributed to the CityLearn environment. I found and proposed a fix for an implementation error that meant that the environment would recompute the entire episode history at every time step t. My fix[1] instead reuses the result of the preceding time step $t-1$, which changes the per-step complexity from $O(t^2)$ to $O(1)$, roughly leading to a 100x-speedup over the course of an episode. I also contributed to finding a bug[2] in the battery model. These efforts were rewarded with the Community Contribution Prize.

The software environment for all experiments uses Python 3.10.9, PyTorch 1.13.0, openAI gym 0.24.1, and CityLearn 1.3.6.

how do I describe this?

The hardware used was a 2020 Macbook Air M1 for the discretization pre-experiment. Tuning and subsequent evaluation runs used the ML-Cloud Slurm cluster, using CUDA on a single Nvidia GTX 2080 GPU per run. The Slurm job scheduling file is included in the attached code repository.

3.3 Experiments

TODO: 2. Explain the experiment details - Experiment goal: Test UA-DQN against DQN and my baseline in this environment. - Metrics: Number of Episodes until convergence, performance at convergence. - introduce challenge setup and data split used in this experiment

3.3.1 Baseline Rule-Based-Controller

In order to be able to evaluate the performance of the Reinforcement Learning agent, I establish a rule-based controller as baseline. Using insights gained from exploratory data analysis, I construct a simple control policy with the goal of minimizing dollar cost.

The basis of the policy is that prices vary predictably. Throughout every day, electricity price is at one of two levels. From 16:00 to 20:00, it is substantially higher than during the rest of the day. In parts of the year, the price level also varies between different days of the week, but the daily pattern still

[1] https://github.com/intelligent-environments-lab/CityLearn/pull/23

[2] https://github.com/intelligent-environments-lab/CityLearn/issues/37

applies. Since battery energy losses are very low, it is therefore worth it to buy and store electricity while it's cheap, and avoid having to buy expensive electricity in the afternoon.

In addition to the electrical grid, the agent also has access to a free, but less predictable source of electricity: solar power. The environment does not allow the agent to sell electricity for a profit. This means any produced solar electricity should either be directly used or stored, since any excess is simply wasted. Directly using solar electricity is more efficient than storing and releasing it at a later time. Putting these basic ideas together, I arrive at a policy that first uses available solar electricity to meet demand. It always stores excess solar electricity, and additionally buys just enough cheap electricity in order to fill up the battery for the more expensive afternoon.

This strategy leaves open one question: When exactly should the battery be charged with electricity from the grid? If it is filled too early, the agent is not able to store excess solar electricity generated during the day. If battery charging starts too late, there is not enough time left to fully charge the battery. Therefore, the battery is charged as late as possible. Lastly, if the battery has remaining charge after the period of high prices, the leftover charge is used as needed, ensuring the battery is free in the morning to store any excess solar power. The final policy is described in algorithm 1.

Algorithm 1 The Rule-Based Controller's Policy always stores excess solar power. Additionally, it tries to charge the battery. After that, it tries to satisfy demand from the battery.

```
  $a \leftarrow (\text{solar} - \text{load})/6.4$        ▷ Difference scaled to units of battery capacity
  if $11 \leq \text{hour} \leq 15$ then
      $a \leftarrow \max(0.24, a)$                               ▷ Slowly charge battery
  end if
Ensure: $-1 \leq a \leq +1$
  return $a$
```

This hand-engineered policy is not perfectly optimized. There are certain insights it does not make use of. It does not directly minimize carbon emissions, though the overall reduced demand for grid electricity leads to a reduction in emissions. The policy also does not incorporate the change in price between weekdays. There are some days on which there is so little excess demand during the day that it would not be worth charging the battery. Finally, the policy does not make use of weather forecasts, which predict solar electricity production.

Overall, the policy serves as a useful benchmark of what a thoughtful human can do when knowing some of the system's dynamics.

Table 3.2: Hyperparameters, their tuning ranges or untuned values.

Hyperparameter	Value
Learning Rate	Tuned: $(0.1, 0.07, 0.03, 0.01, \ldots, 0.00001)$
Batch Size	Tuned: $(1, 2, 4, 8, \ldots, 256)$
Adam's ϵ	Tuned: $(1 \times 10^{-1}, 1 \times 10^{-2}, \ldots, 1 \times 10^{-9})$
Target Network Update Frequency	Tuned: $(4, 8, 16, 32)$
Replay Buffer Size	10,000
Discount Rate γ	0.99
Network Weight scale	$\sqrt{2}$
ϵ-greedy DQN: ϵ	Decay from 0.1 to 0.02 over 1,000 steps
UA-DQN: Quantile Huber Loss κ	10
UA-DQN: aleatoric factor	0
UA-DQN: epistemic factor	1

3.3.2 Discretization

UA-DQN and DQN require a discrete action space. In CityLearn, the action is a continuous number between -1 (releasing energy) and +1(storing energy). The goal of this pre-experiment is to determine a suitable subdivision of the continuous action space. A larger discrete action space means the algorithm learns slower, but a smaller discrete action space means the algorithm can't act as precisely, capping the possible performance. The goal therefore is to find the smallest number of subdivisions that does not incur a significant performance penalty.

In order to measure the importance of different subdivisions, the hand-engineered RBC is run on versions of the environment with differently discretized action spaces. Its performance on the discretized action spaces is then contrasted with its performance on the continuous case. This experiment uses the full 2022 CityLearn challenge public dataset of 5 buildings and one year.

3.3.3 Hyperparameter Tuning

All tested Deep Reinforcement Learning agents and the optimizer, Adam, expose hyperparameters that need to be tuned for optimal performance. Adam's tuned hyperparameters are the learning rate and the parameter ϵ. I also tuned the batch size and the update frequency of the target networks. All other hyperparameters I set to default values as noted in table 3.2.

The process of tuning was to randomly sample 200 combinations of hyperparameters from the sets given in table 3.2 for each DQN variant, and 100 for UA-DQN. The DQN variants ran for 100 episodes, UA-DQN ran for 50 episodes. All runs used the same random seed. Data used for this experiment was the whole year of building 1.

The performance measure for this experiment was the collected total reward in the last episode.

3.3.4 Comparison of Tuned Algorithms

For each algorithm, I selected the best run of hyperparameters and repeated the experiment with 10 different random seeds, all else equal.

Chapter 4

Results

Incorporate Nicole's feedback! (see slack DMs)

4.1 Rule-Based Controller

- How does it perform? - On how many days does it needlessly fill the battery?

4.2 Discretization

In the pre-experiment on the effect of discretization, the rule-based agent performs comparably to the continuous case when discretization resolution is high, and worse on coarse resolutions. The coarsest resolution that performs similarly well as the continuous case is the subdivision into 8 or 9 possible actions.

4.3 Hyperparameter Tuning

All 200 runs of ϵ-greedy DQN completed successfully. 10/100 runs of UA-DQN, all runs with a batch size of 1, failed. 18/200 runs of DQN-softmax failed, all of which share a batch size of 8. However, 17 other runs with a batch size of 8 completed successfully.

- How much computational power did the tuning take?
- if I can find it. If not, just omit.

Table 4.1 shows the results of the tuning process for each algorithm. For each algorithm, the learning rate had the most significant correlation with final performance, followed by the batch size and the target network update frequency. The ϵ parameter of the optimizer Adam showed a large effect only for ϵ-greedy DQN.

Figure 4.2 shows a histogram of the final performance of all successful tuning runs. Both DQN algorithms show a peak at around -5000, which cor-

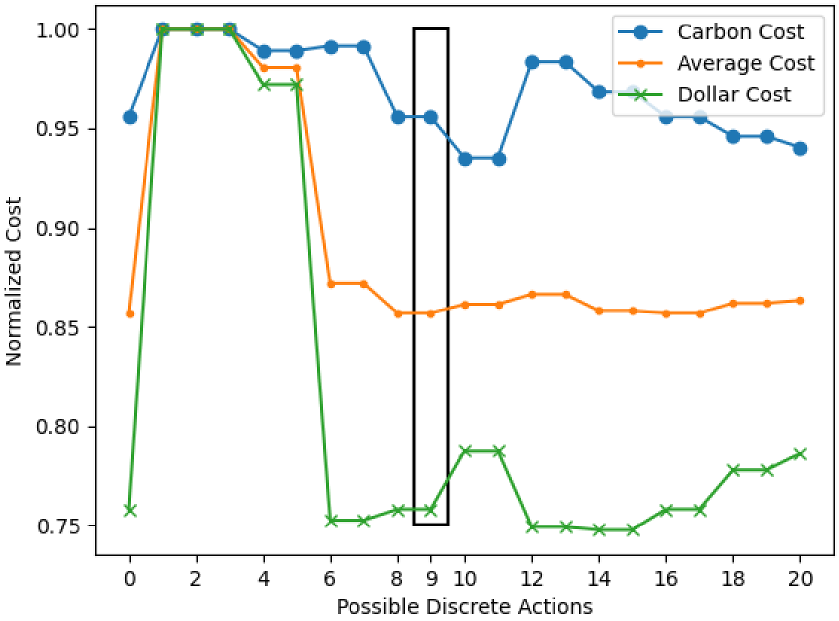


Figure 4.1: This graph shows the effect of the discretization resolution on the rule-based control algorithm. The coarsest resolution that does not significantly impact performance and includes the zero action is 9. In this graph, 0 possible actions means no discretization is applied.

responds to strategies that make no or very little use of the battery. The UA-DQN algorithm learns a superior strategy for more hyperparameters.

4.4 Comparison of Tuned Algorithms

Figure 4.3 shows the episode rewards of the selected hyperparameters for each algorithm during training. Tuned UA-DQN converges faster than the other algorithms, and it converges to a better mean performance, as stated in table 4.2. The hand-engineered rule-based agent outperforms the tuned reinforcement learning algorithms on both metrics.

When repeated with 10 different seeds, a single run of DQN-softmax failed, compared to no failures from the other algorithms.

To illustrate the difference in exploration between the algorithms, figure 4.4 shows the fraction of selected non-greedy actions per episode. All algorithms start out exploring more and then gradually decrease their exploration rate. DQN-softmax and UA-DQN explore more than ϵ-greedy DQN, which quickly reaches an exploration rate of $\epsilon = 0.02$. UA-DQN keeps exploring more than

Table 4.1: This Table shows the correlation between tuned hyperparameters and algorithm performance for successful runs.

Algorithm	Hyperparameter	Value for Best Run	Correlation with final score
UA-DQN	Learning Rate	3e-4	**-0.47**
	Batch Size	128	0.28
	Adam's ϵ	1e-07	-0.03
	Target Network Update Frequency	4	-0.11
DQN-softmax	Learning Rate	3e-4	**-0.36**
	Batch Size	128	-0.18
	Adam's ϵ	1e-05	-0.02
	Target Network Update Frequency	4	0.16
DQN-ϵ-greedy	Learning Rate	7e-05	**-0.33**
	Batch Size	4	-0.19
	Adam's ϵ	1e-08	0.10
	Target Network Update Frequency	16	0.14

the other algorithms.

Table 4.2: This table shows the mean performance of tuned algorithms when evaluated using their respective action selection policy for one episode on Building 1.

Agent	Dollar Cost	Carbon Emission	Average
Control (No Action)	1	1	**1**
DQN-Softmax	0.83	0.93	**0.88**
DQN ϵ-greedy	0.82	0.93	**0.88**
UA-DQN	0.82	0.91	**0.87**
Discrete Rule-Based	0.80	0.88	**0.84**

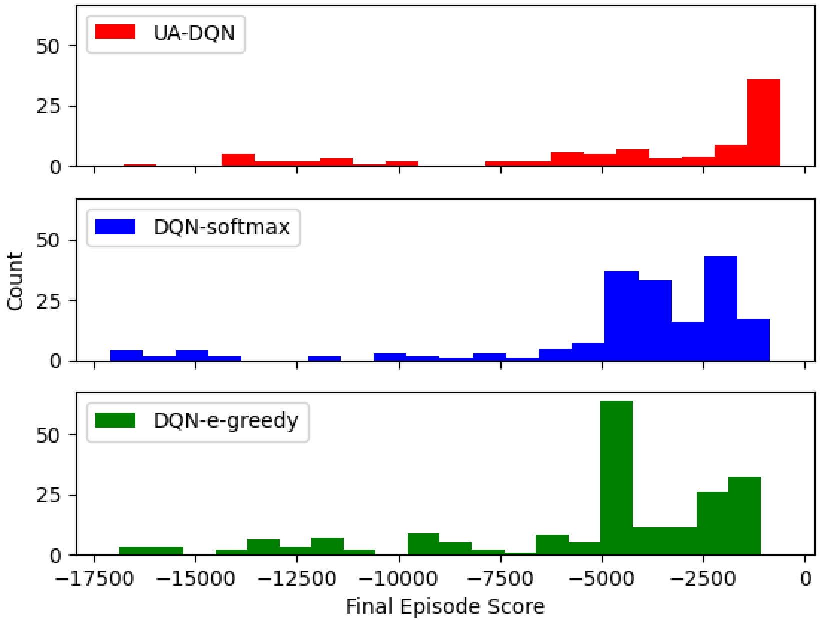


Figure 4.2: This histogram shows the final episode reward of all successful tuning runs.

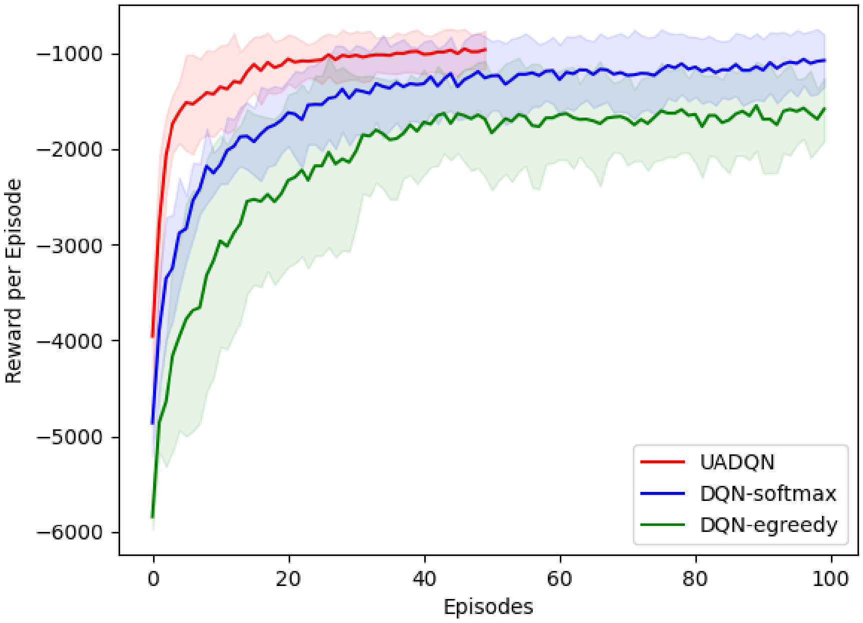


Figure 4.3: This graphic shows the performance during training of the three tuned algorithms. The shaded area shows the standard deviation over 10 runs with different random seeds. Tuned UA-DQN converges after fewer episodes than either DQN variant. UA-DQN was only trained for 50 episodes due to the added computational complexity of the algorithm.

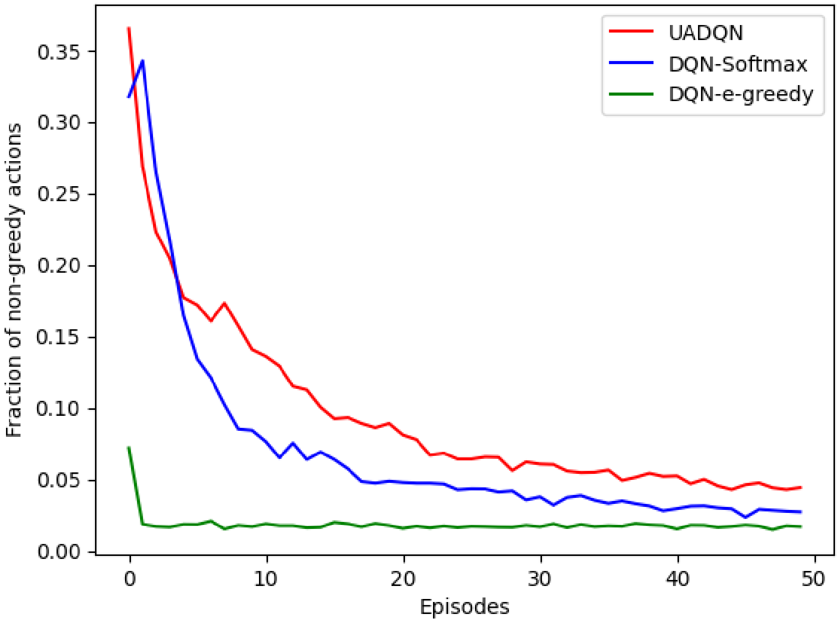


Figure 4.4: This graphic shows the fraction of selected actions that do not correspond to the highest expected value. It highlights the different action selection strategies employed by the different algorithms. UA-DQN keeps exploring for longer than the other strategies.

Chapter 5

Discussion

Of course very important! You need to discuss the informatics as well as econ part of your thesis topic.

Take your time for writing the discussion, besides the introduction chapter it is the most important chapter of your thesis. Also do not subsection the discussion too heavily.
At least 5 pages,
Outlook can become an extra chapter.

Outline/structure for this chapter: TODO: Fill with content
- restate research question: Does an uncertainty-aware algorithm outperform other methods on this problem? - summarize key findings (answer question) - On this problem, UA-DQN learns faster than other methods. –¿ It explores more efficiently. - Its final performance is not significantly better than others (compute p value maybe, do t-test. Does that make sense?) - For a simple system, a tuned rule-based policy can outperform RL methods. - UA-DQN takes more computational resources. - explain and interpret results in more detail - relate back to other literature - support conclusions with data from results - ... - Discuss limitations - Internal Validity: Does the experiment support the conclusion? - Experiment was not thorough enough - Absence of good performance does not mean good performance is impossible, maybe I just did it wrong - Data was not fully used - External Validity: Does the conclusion generalize to real life? - Model limitations: Does CityLearn represent the real life application? - The real life application of such a system depends on technical and regulatory details - Data is real life, but other data might be available. - A real life application could have different goals than only price and CO2, like coordination goals and communication requirements. - - Is the question important? - what should such a system even be judged on in application? - Further Research - Application in a real life system -

mention that e-greedy epsilon was not tuned, but it definitely should have been!

- Tuning: Initialization might have been different for different algorithms
- What would a good setting for the risk-aversion parameter even be?

Known Limitations of the RL environment

- data: Weather forecasts are not real forecasts - energy model: Battery charging speed is symmetric - market model: No selling the electricity - Coordination goals: Were introduced later in the challenge, not used here. - Only single building.

CityLearn as a model for building-based demand response: - limitations: modelling errors/simplifications and how much they matter - CityLearn was built to assess the general usefulness of RL for DR, not for my specific question. - Battery efficiency - Weather Forecasts are perfect oracles - Only available action is battery charging/discharging. - Other processes are assumed to be fixed (at least in 2022 challenge). Automatic Washing Machine starting. - Human behavior can not be influenced (models humans as inflexible) - Hourly control instead of continuous -¿ Makes perfect control more difficult - Energy can not be sold to the grid, not even to the microgrid. -¿ unrealistic - 2022 challenge: initially did not encourage cooperation. - Strengths: For what tasks is this environment adequate? - Makes it possible to apply RL to DR without expensive computational overload* - Test Multi-Agent behavior and cooperation in the DR context, as opposed to only single-building frameworks - Enables comparison of different approaches as a benchmark

Chapter 6

Conclusion and Outlook

- 1 page - summarize again what your paper did, but now emphasize more the results, and comparisons - write conclusions that can be drawn from the results found and the discussion presented in the paper - future work (be very brief, explain what, but not much how)

Bibliography

CityLearn v1.0: An OpenAI Gym Environment for Demand Response with Deep Reinforcement Learning. *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, 2019. doi: 10.1145/3360322.3360998. 3.2

Han Li, Zhe Wang, Tianzhen Hong, and Mary Ann Piette. Energy flexibility of residential buildings: A systematic review of characterization and quantification methods and applications. *Advances in Applied Energy*, 3:100054, August 2021. ISSN 2666-7924. doi: 10.1016/j.adapen.2021.100054. 2.2

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, February 2015. ISSN 1476-4687. doi: 10.1038/nature14236. 2.3.2

Ram Narayanamurthy, Rachna Handa, Nick Tumilowicz, C R Herro, and Sunil Shah. Grid Integration of Zero Net Energy Communities. page 12. 3.2.1

Hannah Ritchie, Max Roser, and Pablo Rosado. Energy. *Our World in Data*, October 2022. 2.1

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning Series. The MIT Press, Cambridge, Massachusetts, second edition edition, 2018. ISBN 978-0-262-03924-6. 2.3.1, 2.3.1

Selbständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbständig und nur mit den angegebenen Hilfsmitteln angefertigt habe und dass alle Stellen, die dem Wortlaut oder dem Sinne nach anderen Werken entnommen sind, durch Angaben von Quellen als Entlehnung kenntlich gemacht worden sind. Diese Masterarbeit wurde in gleicher oder ähnlicher Form in keinem anderen Studiengang als Prüfungsleistung vorgelegt.

Ort, Datum	Unterschrift