Active Reinforcement Learning for Robust Building Control

Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

Master in Artificial Intelligence and Robotics

Course Reinforcement Learning
Professor Roberto Capobianco
TA Michela Proietti
Student Luigi Gallo (1895146)

RL for Optimal HVAC Control

HVAC (Heating, Ventilation and Air Conditioning) systems are critical components of building infrastructure, consuming significant **energy** resources.

Optimizing their **control** presents challenges due to :

- ***Dynamic*** environmental conditions
- ***Diverse*** building *usage* ***patterns***
- Occupant ***comfort*** requirements

Reinforcement Learning offers a promising approach by enabling the HVAC system to **learn optimal control** strategies through interaction with its environment, ultimately leading to *adaptive* and *efficient* operation.

RL for Optimal HVAC Control

Several studies have explored the **application** of Reinforcement Learning (RL) techniques for optimizing **HVAC** control in buildings.

While these approaches have shown **promise**, they often face challenges in **adapting** to sudden **environmental changes**, such as *extreme* weather conditions.

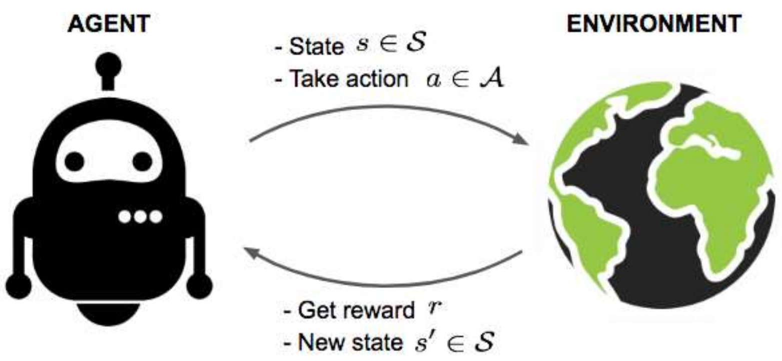
Is this really a problem?

Climate Change

The answer is… **YES, it is**

Active RL for Robust Building Control

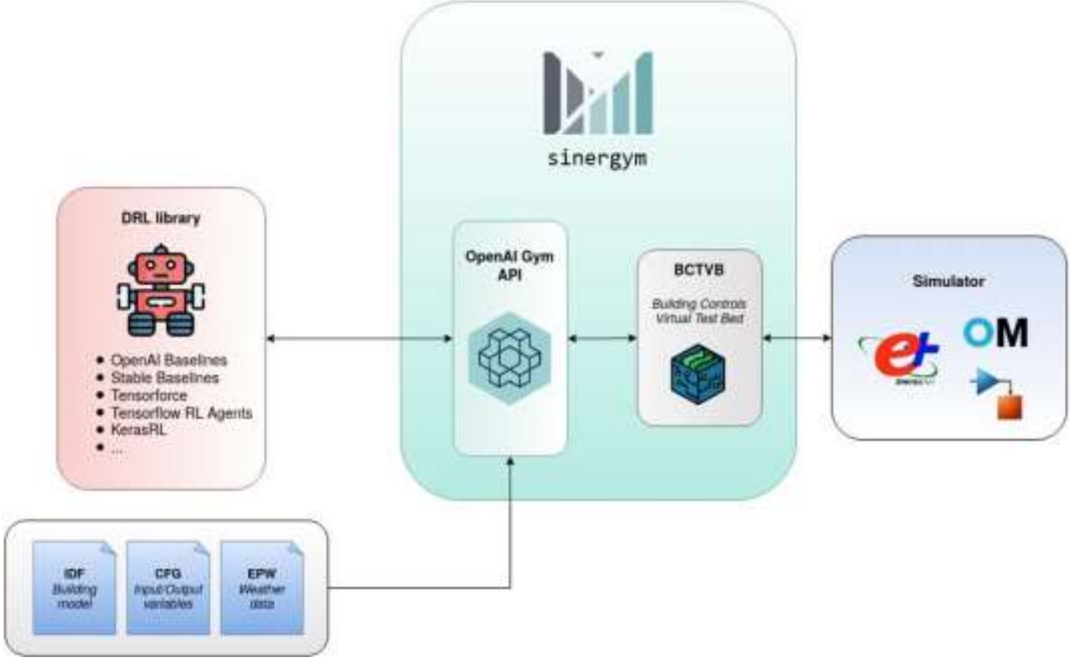
- RL **agents** can get too used to their training and **struggle** with **new situations**.
- Unsupervised Environment Design (**UED**) **helps** by training these agents in *specially* **crafted environments**.
- Not all UED approaches fit well when we need the agent to do well in one specific environment, especially under different conditions like normal or extreme weather.
- In this work, I introduce a new UED method called **ActivePLR**. It smartly creates training scenarios that challenge the RL agent just enough, focusing on **maintaining** a building's energy **efficiency** and **comfort** during *various* **weather** conditions in a chosen environment.



The Environment

Sinergym is a powerful tool for modeling building **environments** and simulating HVAC systems' behavior within those environments.

It serves as an interface to **EnergyPlus**, a widely used building energy **simulation engine**, providing a user-friendly environment for HVAC control experimentation and analysis.

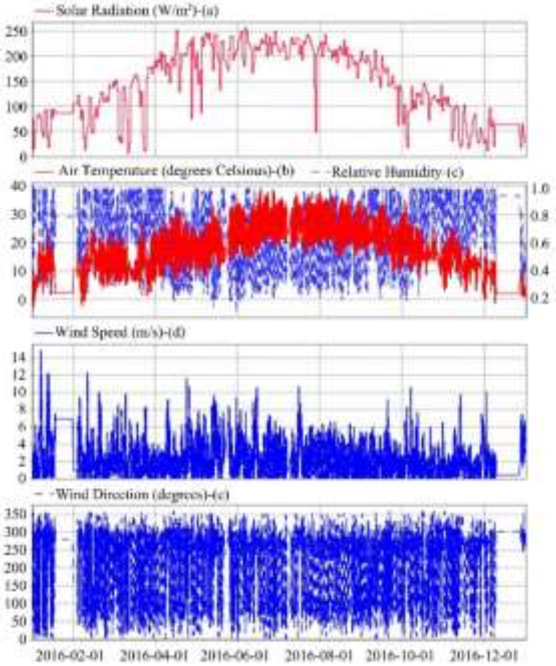


Unsupervised Environment Design

To make the agent more robust to changes in the environment, an environment **wrapper** must be created.

By default, synergym varies only the external temperature based on (**mu**, **sigma**, **theta**) specified. With this wrapper, you can now specify parameters of (mu, sigma, theta) for **5** environment **factors**:

- Outdoor temperature (**drybulb**)
- Relative humidity (**relhum**)
- Wind direction (**winddir**)
- Wind speed (**windspd**)
- Solar irradiance (**dirnorrad**)



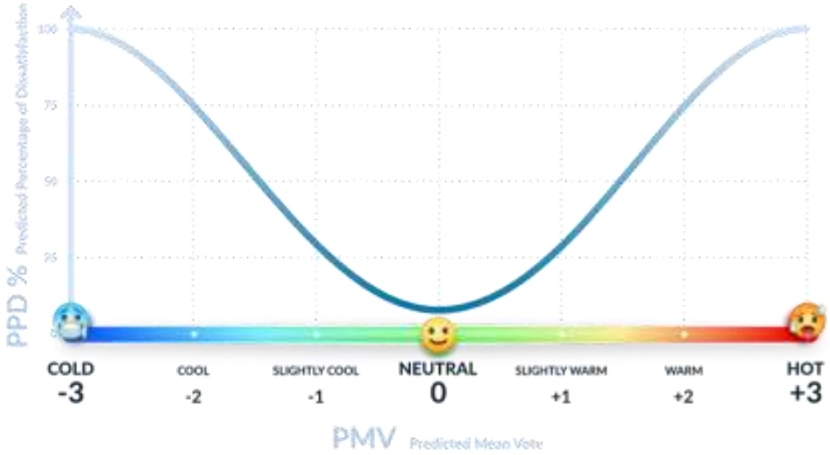
In order to make these changes, it was necessary to **override** default sinergym functions.

The Reward Function

Linear **reward** function using **Fanger** PPD thermal comfort is also created. This considers the **energy consumption** and the **PPD** (Percentage of People Dissatisfied) thermal comfort metric, as well as occupancy:

$$R_t = -\rho \cdot \lambda_E \cdot P_t - (1-\rho) \cdot \lambda_p \cdot PPD_t \;\cdot\; \mathbb{I}_{occupancy_t > 0} \cdot \mathbb{I}_{PPD_t > 20}$$

Sinergym offers the ability to define *custom reward* functions very easily. This reward is derived from a standard **linear reward** by modifying the **comfort** factor only.



RAY

Ray is an open-source framework designed to **simplify** and **scale** up machine learning and other intensive computation tasks. It provides a simple, universal API to build distributed applications.
RLlib is a reinforcement learning library built on top of Ray.

- **Flexible**: easy integration with various *environments*.
- **Scalable**: Designed to scale from a single CPU to large clusters with minimal effort.
- **Comprehensive**: supports a *wide variety* of RL algorithms, including ***PPO***, DQN, and A3C.

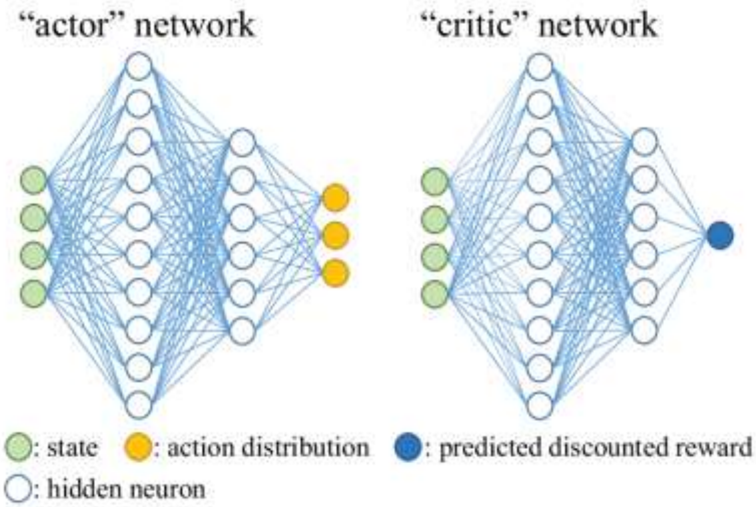
Chosen for its flexibility with the goal of testing the modeled environment.

The Agent

Proximal Policy Optimization (**PPO**) is a type of **policy gradient method** for training deep reinforcement learning algorithms.

PPO achieves this through a novel **objective function** that balances the *exploration-exploitation* trade-off more effectively than its predecessors.

This method **limits** the size of policy updates, reducing the likelihood of performance degradation due to **large**, destabilizing **updates**.

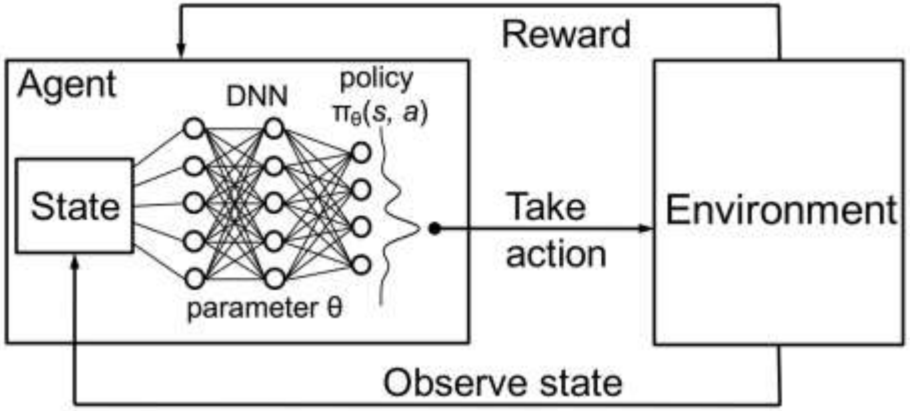


Training

Agent training took place in each test over a total of **500** training **episodes**. The **goal** of the training was to **understand** what **factors** were **significant** in improving agent performance.

I started from a **baseline** using the **default** configurations of RLlib, and made changes to the training setup with the goal of achieving improvements.

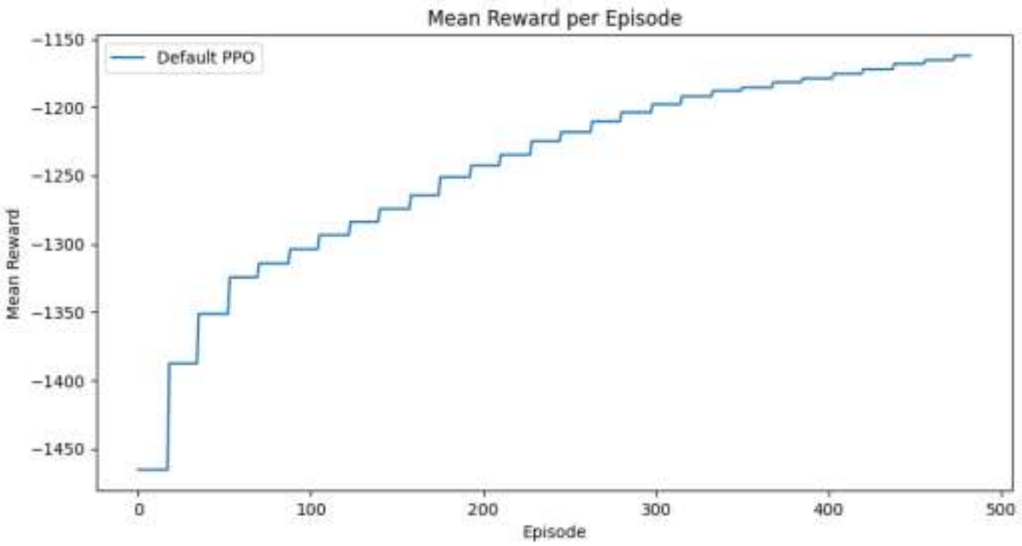
To evaluate an agent's performance, the **mean reward** obtained at each **episode** during training was considered as a metric.



Default PPO

With the goal of setting a **baseline** from which to start, the first tests performed were with the **default** configuration of **RLlib**'s PPO.

Some of these default parameters include the use of the **hyperbolic tangent** as a trigger function for fully connected layers, and a discount factor **gamma** of 0.99.

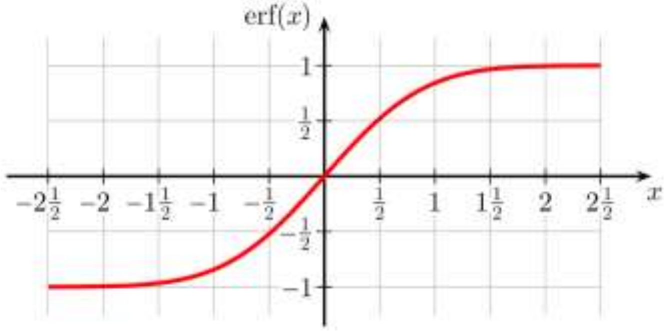


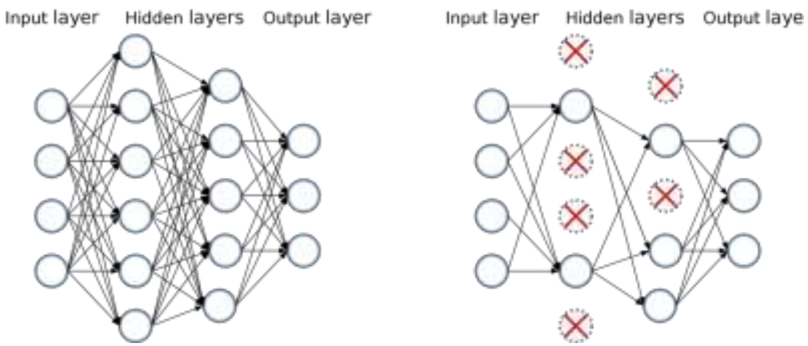
Activation function

One of the factors that influenced performance significantly was changing the **activation function** of the PPO.

By default, this is a simple **hyperbolic tangent**, but adding a **dropout** to it (**p=0.3**) resulted in improvements.

This is because the *dropout* acts as a **regularizer**: with a certain **probability** p, some of the **information** at each layer is **masked**, forcing the model to rely on only a reduced part of the network, thus **improving generalization** performance.

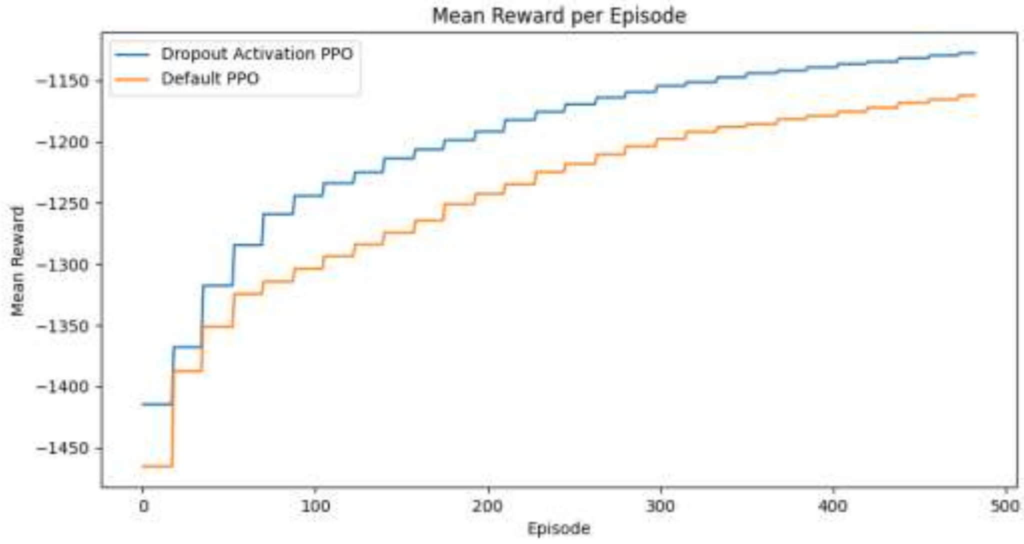




Activation function

After changing the default activation function, one can see a slight **improvement** of the agent on the rewards obtained.

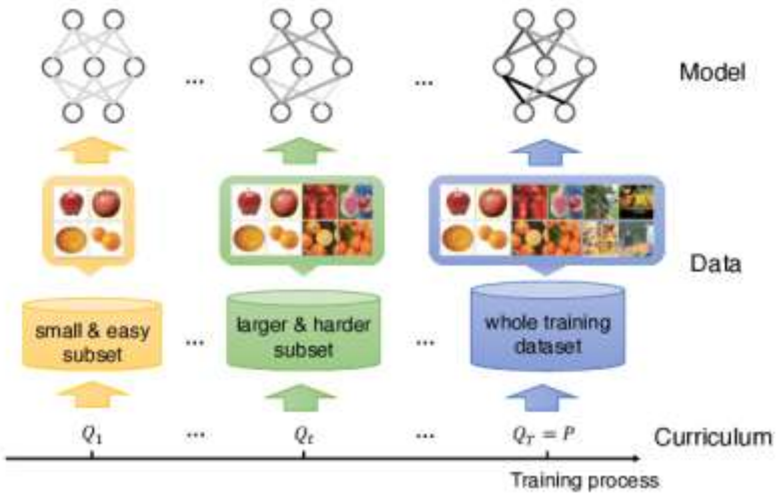
This is because as the surrounding environment is more **variable**, adding a **dropout** increases the **generalization** ability of the model.



Curriculum Learning

Another strategy I decided to test is **Curriculum Learning**: start with an easier task and **gradually increase** the **difficulty**.

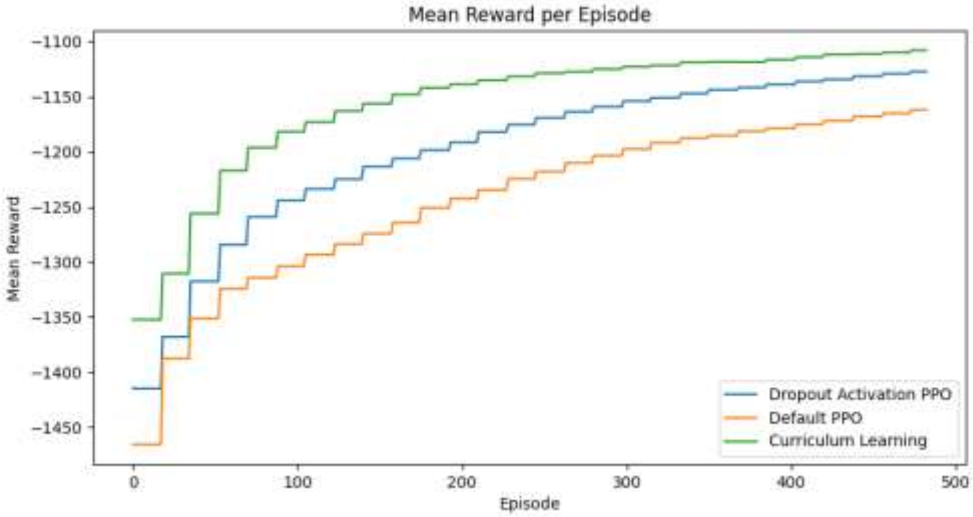
As described earlier, the defined environment allows 5 environmental factors to vary significantly. Instead of starting the training in a more complex environment in which all variables are active, a new **variable** is **introduced at** a **fixed cadence**, performing for *75%* of the training a gradual learning, and for *25%* a learning at maximum difficulty.



Curriculum Learning

In the **initial stages**, when the environment is less *complex*, it's easier for the agent to **understand** the relationship between its **actions** and the resulting **rewards**.

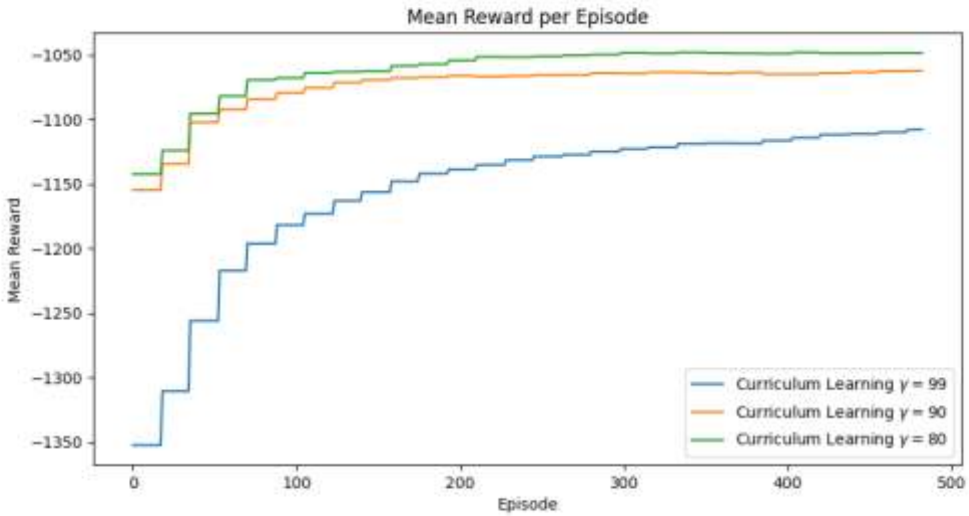
Incremental learning **encourages** the development of a more **robust policy**, and by the time the agent is exposed to the full complexity of the environment, it has already learned to **adjust** its strategy in **response** to changes in **one variable** at a **time**.



Gamma

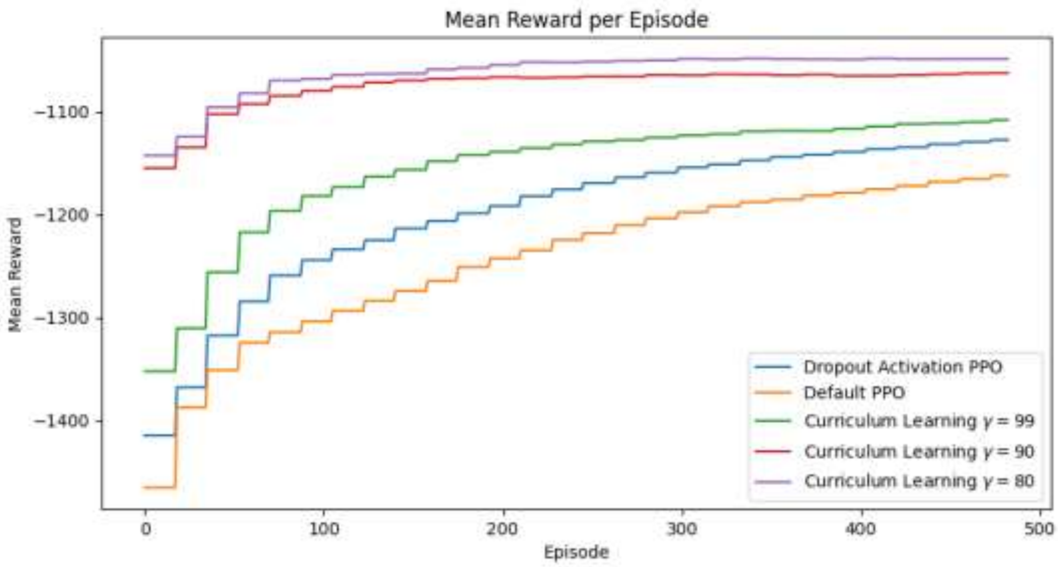
The **discount factor** gamma, that determines the **importance** of **future rewards** in the total expected rewards for an agent's actions.

After testing various values, the optimal one is **0.8**. This indicates that **immediate** or short-term **decisions** have a **more significant** impact on performance than long-term strategies.
That’s because the **benefits** of adjusting HVAC settings might **diminish** over time due to *external factors* like weather changes or occupancy variations.



Improvements overview

Through strategic modifications, including **dropout** regularization, **Curriculum Learning**, and **gamma** adjustment, our RL agent now achieves **higher rewards**, indicating enhanced adaptability and efficiency in building control.



Thanks for your attention