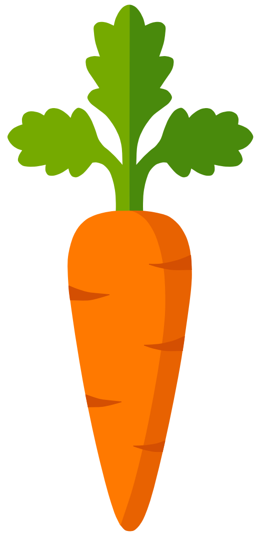
SAPIENZA
UNIVERSITÀ DI ROMA

# Reinforcement Learning Final Project



## CropGym Intercropping Extension

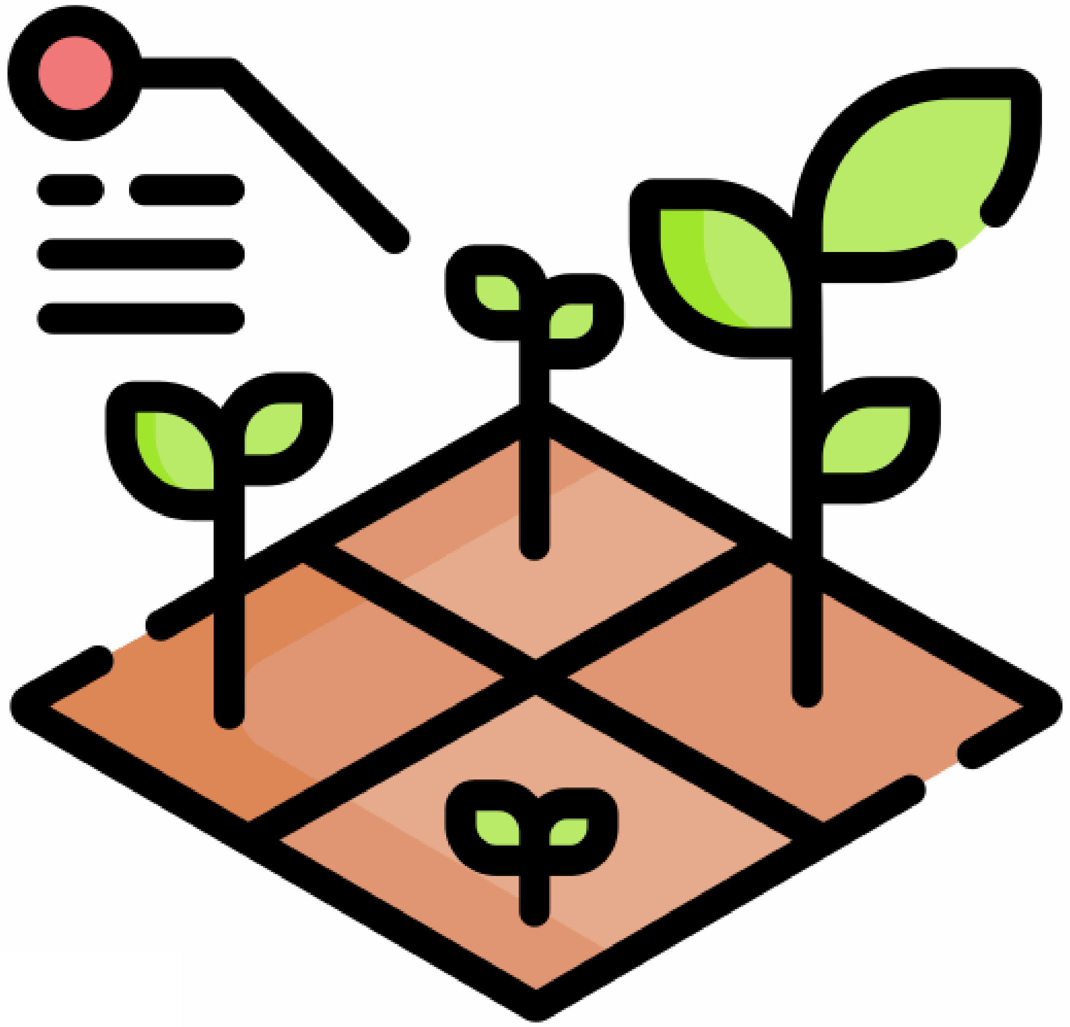

**Leonardo Sandri & Federico Matarante**
*2137374 & 2133034*

sandri.2137374@studenti.uniroma1.it
matarante.2133034@studenti.uniroma.it

SAPIENZA
UNIVERSITÀ DI ROMA

# Index

# Introduction



### Inter-cropping:

Growing multiple crops together, improving sustainability through better resource use and biodiversity.

# Introduction

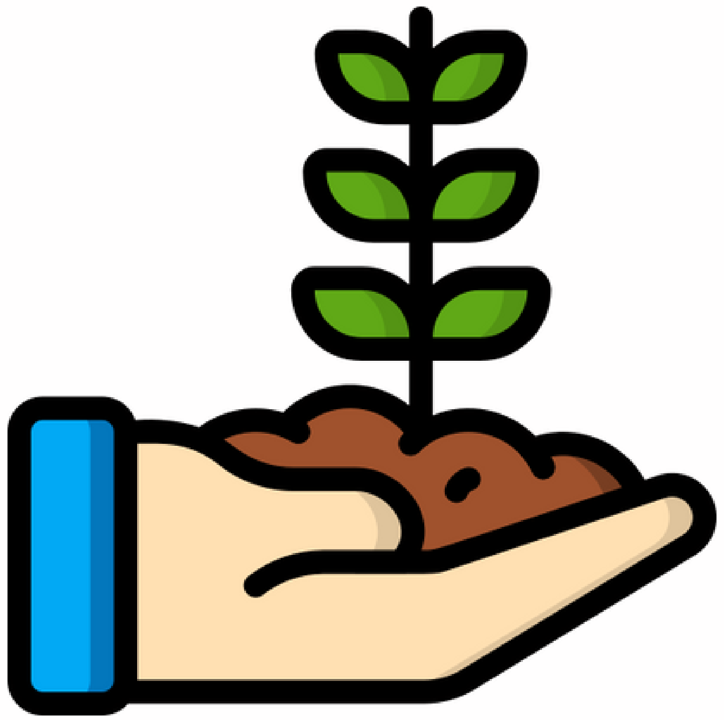
**Aim:**

Contributing in AI application on sustainable farming practices

**Idea:**

Building from scratch an extension of a pre-existing environment in order to provide to future researchers a tool for simulating inter-cropping techniques

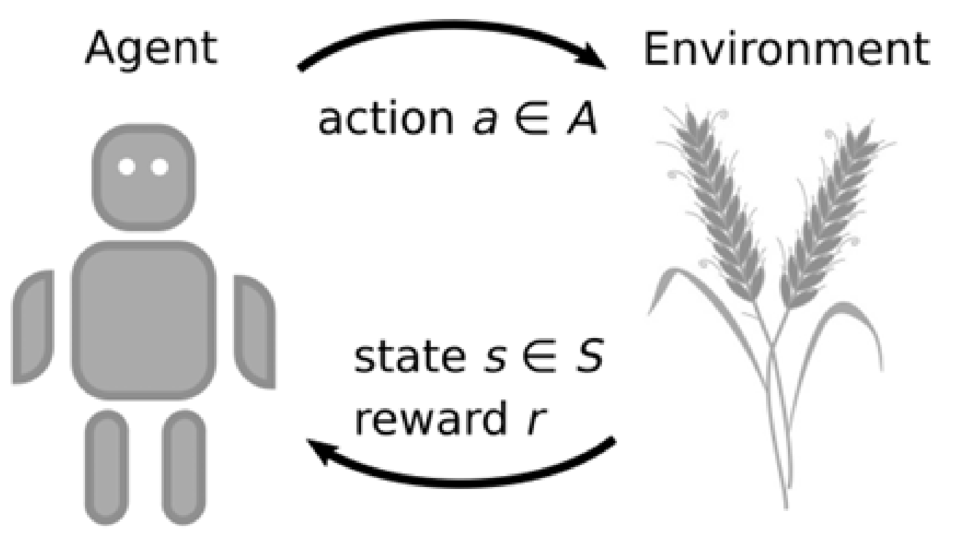
# Environment



**CropGym:**

RL environment based on the Lintul3 engine for fertilizer optimization on a nitrogen-limited environment.
Used as a base for custom environment.

# Environment




## CropGym:

- Action space: quantity of fertilizer to use every 7 days
- Reward: tradeoff between account fertilizer used and crop growth compared to a baseline
- Observation space: crop, soil and weather variables.

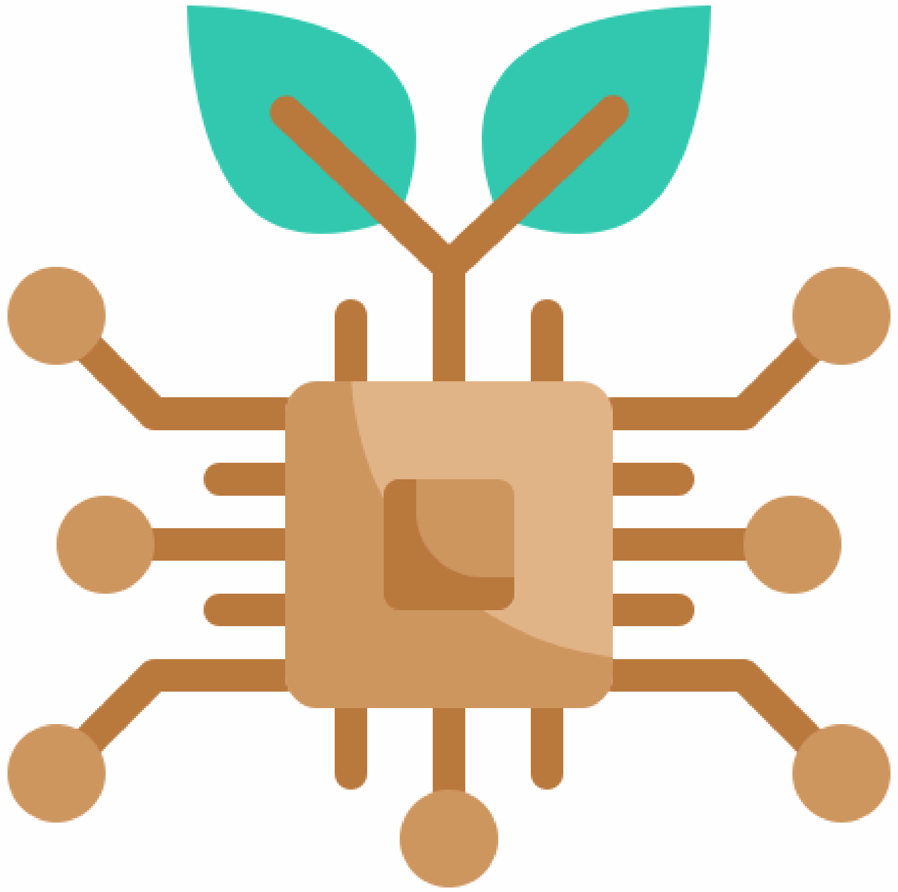
*Action Space* →

$$A = \{20k\frac{\text{kg}}{\text{ha}} \mid k \in \{0, 1, 2, ..., 6\}\}$$

*Reward Formula* →

$$r_t = m_{SO,t} - m_{SO,t-1} - (m^*_{SO,t} - m^*_{SO,t-1}) - \beta m_{fert,t}$$

# Environment



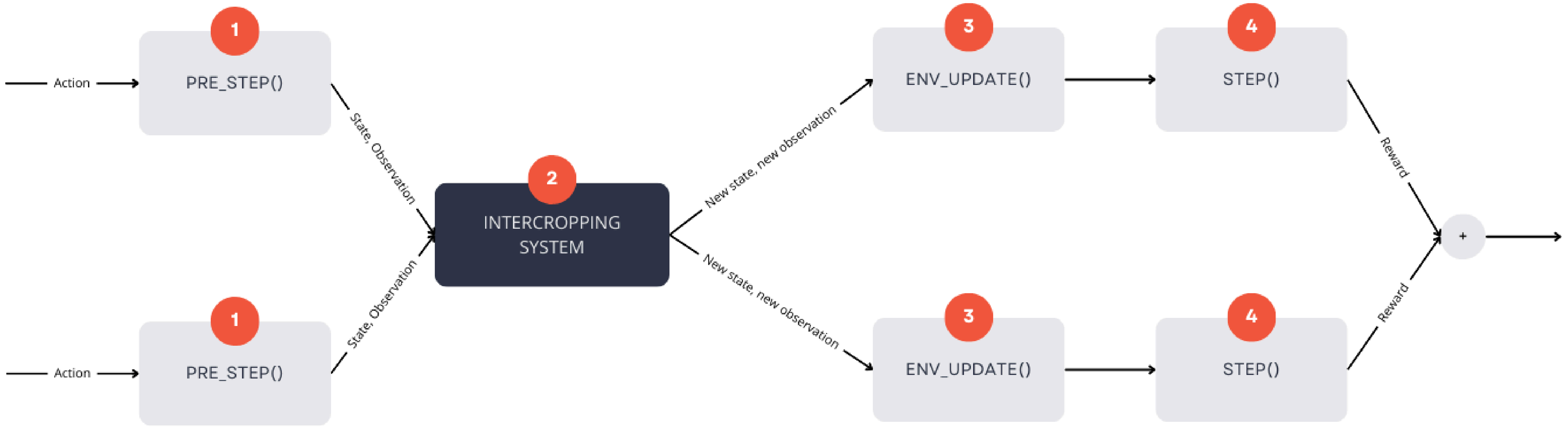
## Our extension:

InterCropGym combines two CropGym instances, applying intercropping techniques. The agent's nitrogen choice is used for both instances, and rewards are summed.

**Formulas and data used have not been validated by agronomists**

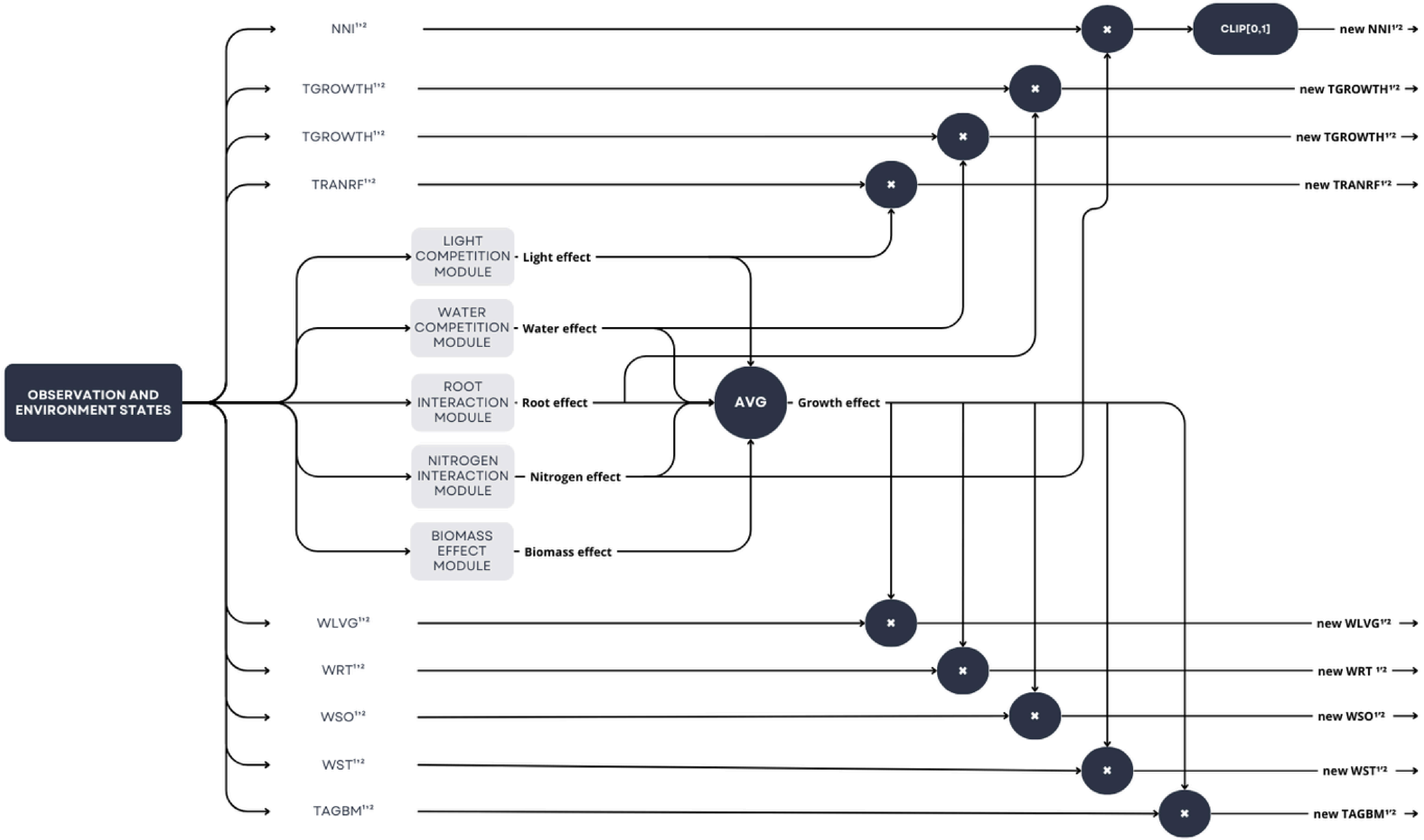
**Disclaimer**

# Environment



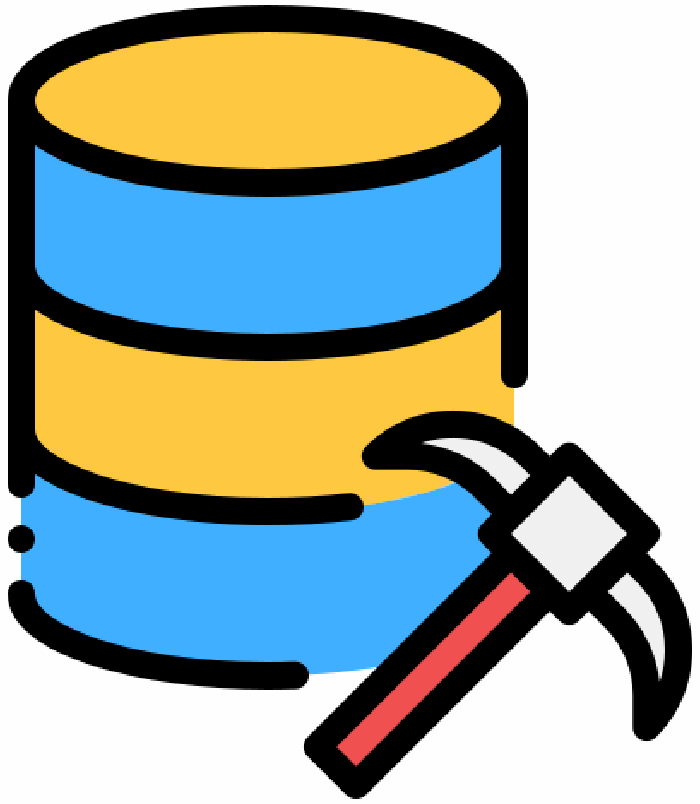

***InterGymCrop structure***

# Environment




*Intercropping System*

# Dataset



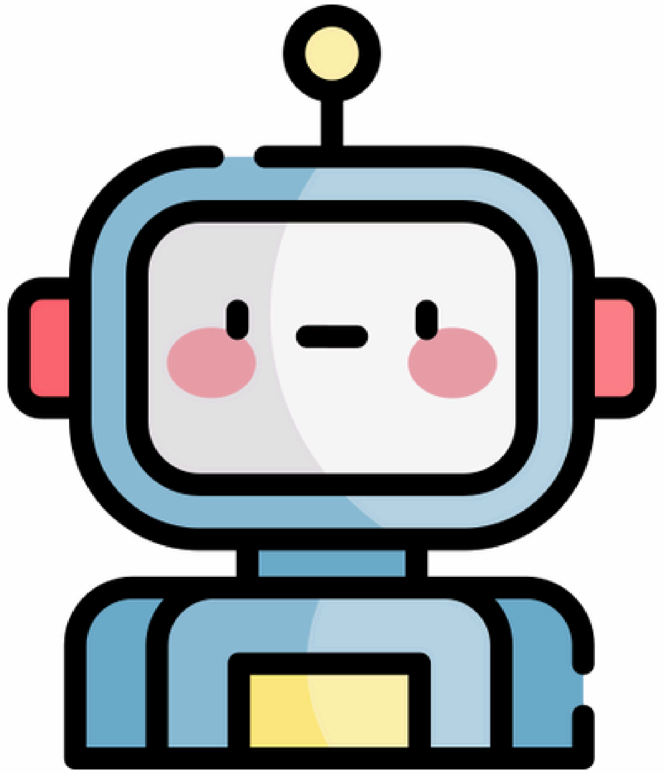
DSSAT / APSIM / WOFOST / PCSE

Concatenation & merging

Generation

# Agents

| Parameter | | Value |
|---|---|---|
| Optimizer | Learning Rate | 5e-4 |
| | Optimizer | Adam |
| | Max gradient norm | 0.5 |
| Network | Actor hidden sizes | [256, 256] |
| | Critic hidden sizes | [64, 64] |
| | Activation | ReLU |
| Algorithm | $\gamma$ | 0.99 |
| | GAE $\lambda$ | 0.95 |
| | Clip_range | 0.2 |
| | Entropy coefficient | 0.05 |
| | Value function coefficient | 0.5 |
| Training Process | Buffer size | 4096 |
| | Batch Size | 512 |
| | Epochs per update | 10 |
| | Eval Frequency | 25 |
| | Eval episodes | 80 |



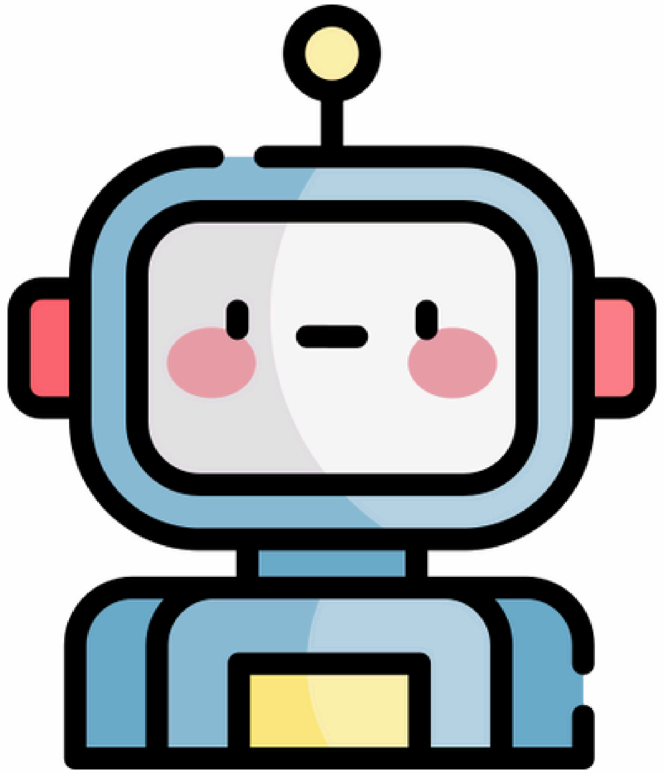
**PPO**

- Actor-Critic architecture
- Clipped Objective Function
- Generalized Advantage Estimation (GAE)

$$\theta_{t+1} = \theta_t + \alpha \nabla_\theta L^{CLIP}(\theta_t)$$

# Agents

| Parameter | | Value |
|---|---|---|
| Optimizer | Learning Rate | 5e-4 |
| | Optimizer | Adam |
| | Max gradient norm | 0.5 |
| Network | Actor hidden sizes | [256, 256] |
| | Critic hidden sizes | [64, 64] |
| | Activation | ReLU |
| Algorithm | $\gamma$ | 0.99 |
| | GAE $\lambda$ | 0.95 |
| | Clip_range | 0.2 |
| | Entropy coefficient | 0.05 |
| | Value function coefficient | 0.5 |
| Training Process | Buffer size | 4096 |
| | Batch Size | 512 |
| | Epochs per update | 10 |
| | Eval Frequency | 25 |
| | Eval episodes | 80 |



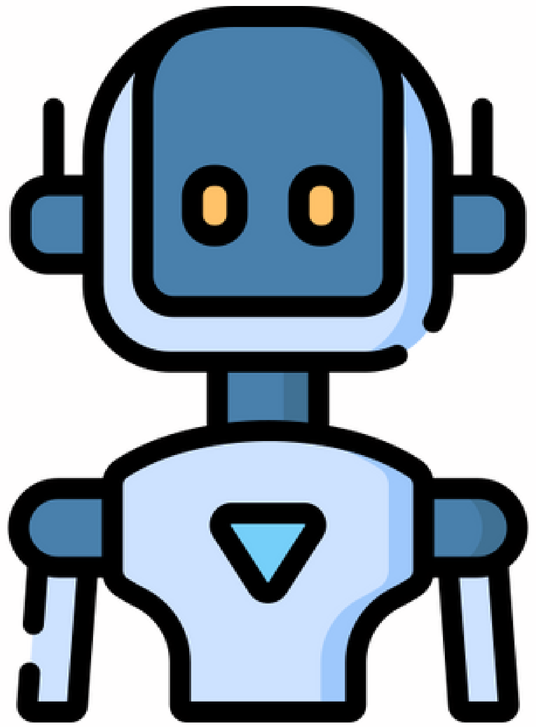
## PPO

Reaching stable convergence and consistent training required time and specific techniques:

- Value function clipping
- State normalization

# Agents

| Parameter | | Value |
|---|---|---|
| Optimizer | Learning Rate | 1e-4 |
| | Optimizer | Adam |
| | Max gradient norm | 20.0 |
| Training Process | Batch Size | 32 |
| | Epochs | 660 |
| | Eval Frequency | 25 |
| | Eval episodes | 80 |
| Algorithm | $\gamma$ | 0.99 |
| | $Q_t$ Update Frequency | 40 |
| Exploration | $\epsilon_0$ | 1.0 |
| | $\epsilon_f$ | 0.05 |
| | $decay\_rate$ | 0.9999 |
| Replay Buffer | Capacity | 5000 |



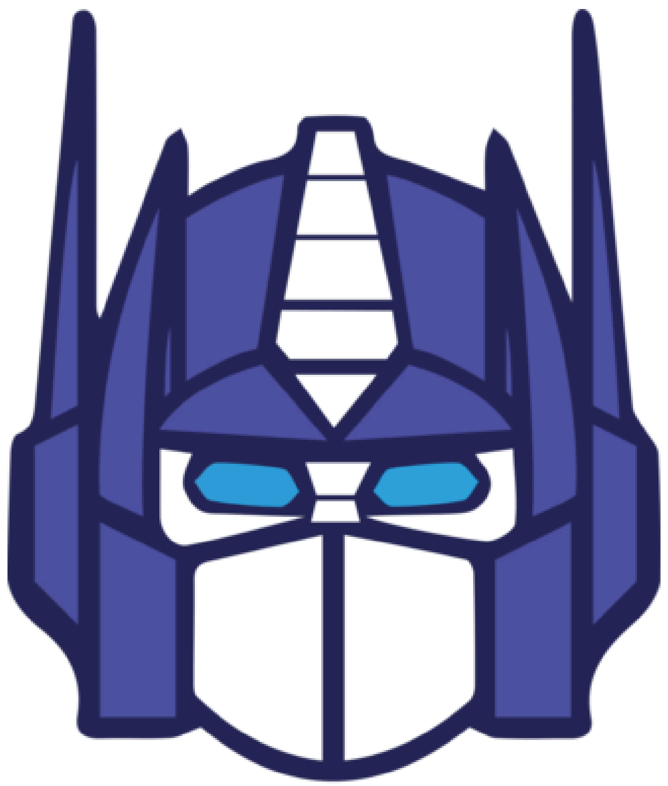
## Double Deep Q-Learning ( DDQN )

- Double Q-Learning
- Temportal Difference Learning
- Replay buffer
- Action selection with online Q-network

*Action Selection:* $a \leftarrow argmax_a(Q_o^\theta(s,a))$

# Agents

| Parameter | | Value |
|---|---|---|
| Optimizer | Learning Rate | 1e-3 |
| | Optimizer | Adam |
| | Max gradient norm | 10.0 |
| Training Process | Batch Size | 256 |
| | Epochs | 660 |
| | Eval Frequency | 25 |
| | Eval episodes | 80 |
| Algorithm | $\gamma$ | 0.99 |
| | $Q_t$ Update Frequency | 5 |
| | $\tau$ | 0.005 |
| Exploration | $\alpha$ | 0.2 |
| | $update\_frequency$ | 1 |
| Replay Buffer | Capacity | 5000 |



## Soft Actor Critic ( SAC )

- Double Q-Learning
- Entropy learning
- Replay buffer
- Actor-critic architecture
- Soft Updates for target Q-network

*TD error:* $\delta = r + \gamma Q_t^{\Theta}(S', A') - (Q_o^{\theta}(s, a) - \xi \log P(A'))$

*Soft Update:* $\Theta \leftarrow (1 - \tau)\Theta + \tau\theta$

# Experimental results

| | PPO | DDQN | SAC |
|---|---|---|---|
| *Reward* | **-657.94** | **-695.0** | **-372.29** |
| *Yield crop 1* | $20.64 \frac{g}{m^2}$ | $20.64 \frac{g}{m^2}$ | $20.64 \frac{g}{m^2}$ |
| *Yield crop 2* | $579.56 \frac{g}{m^2}$ | $579.56 \frac{g}{m^2}$ | $579.52 \frac{g}{m^2}$ |
| *Fertilizer used* | $60.76 \frac{kg}{ha}$ | $64.44 \frac{kg}{ha}$ | $33.04 \frac{kg}{ha}$ |

# Conclusion and future work

Our aim was to create a reference.
Future agronomists may review the work and adjust formulas and data to realistic applications.
Further experiments with model may be done in order to improve performance and obtain real-world approaches for sustainable agriculture.

# References


- **CropGym**: HadovanHasselt,ArthurGuez,andGoogleDeepMind DavidSilver.“DeepReinforcementLearningwithDoubleQ-learning”. In:arXiv:1509.06461v3(2015).
- **PPO**: John Schulman et al. “Proximal Policy Optimization Algorithms”. In: arXiv:1707.06347 (2017).
- MichielG.J.Kallenberget al. “Nitrogenmanagement withreinforcement learningandcropgrowthmodels”. In: EnvironmentalDataScience 2 (2023), e34. DOI: 10.1017/eds.2023.28.
- **SAC**: Tuomas Haarnoja et al. “Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor”. In: arXiv:1801.01290 (2018).
- **DDQN**: Hado van Hasselt, Arthur Guez, and Google DeepMind David Silver. “Deep Reinforcement Learning with Double Q-learning”. In: arXiv:1509.06461v3 (2015).
- **DSSAT**: Phillip D Alderman. DSSAT: A Comprehensive R Interface for the DSSAT Cropping Systems Model, R package version 0.0.9. 2024. DOI: 10 . 5281 / zenodo . 4091381. URL: https : / / CRAN . R -project.org/package=DSSAT
- **APSIM**: Dean P. Holzworth et al. “APSIM – Evolution towards a New Generation of Agricultural Systems Simulation”. In: Environmental Modelling & Software 62 (Dec. 2014), pp. 327–350. DOI: 10 . 1016 / j.envsoft.2014.07.009
- **pcse**: Allard de Wit. PCSE - Python Crop Simulation Environment. Accessed: 2024-02-21. 2024. URL: https://github.com/ajwdewit/pcse
- **WOFOST**: Allard de Wit. PCSE - Python Crop Simulation Environment. Accessed: 2024-02-21. 2024. URL: https://github.com/ajwdewit/pcse
- Balderas et al. (2024): Joseph Balderas, Dong Chen, Yanbo Huang, Li Wang, and Ren-Cang Li. “A Comparative Study of Deep Reinforcement Learning for Crop Production Management.” In: arXiv preprint arXiv:2411.04106 (2024). URL: https://arxiv.org/abs/2411.04106.
- Overweg et al. (2021): Hiske Overweg, Herman N.C. Berghuijs, and Ioannis N. Athanasiadis. “CropGym: a Reinforcement Learning Environment for Crop Management.” In: arXiv preprint arXiv:2104.04326 (2021). URL: https://arxiv.org/abs/2104.04326