Accelerating AI Through Human Knowledge

Teaching to Imitate Experts and Win on the Race Track

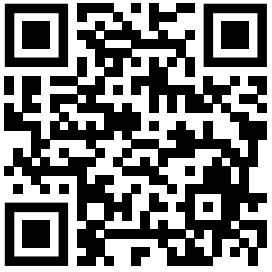
Alexander Buchelt, Tobias Kietreiber

/computer science & security
/fh///
st.pölten

About Us

- Alexander Buchelt
  - Junior Researcher at St. Pölten UAS
  - alexander.buchelt@fhstp.ac.at
- Tobias Kietreiber
  - Junior Researcher at St. Pölten UAS
  - tobias.kietreiber@fhstp.ac.at

Link to the Github Repo



https://github.com/fhstp/MLPragueImitation

Reinforcement Learning

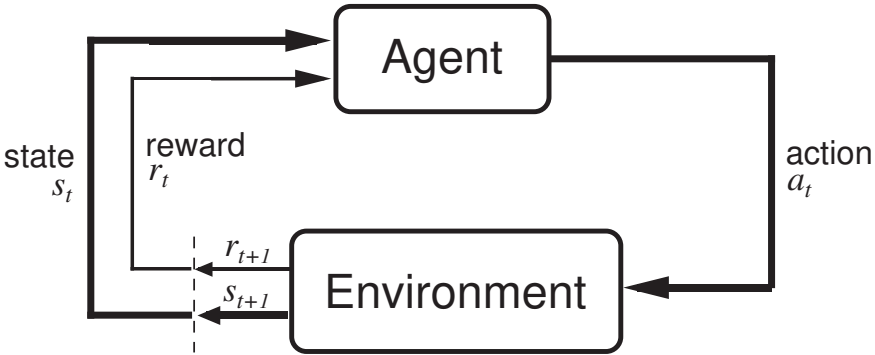
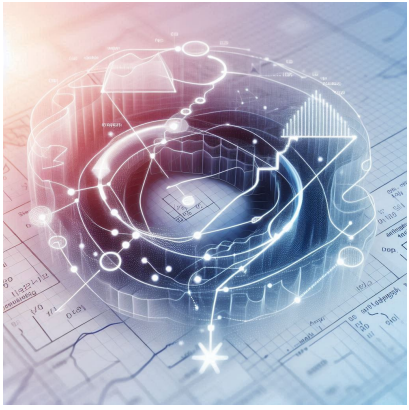


Figure 1: Taken from (Sutton & Barto, 2018)

Overview of the RL Process

- Design a meaningful state
- Make sure all needed information is present in the state!
- Design a good reward signal, enabling the agent to learn
- Choose an RL algorithm fitting the problem



Chess/Shogi/Go (DeepMind, 2018)

- Environment: board + pieces (+ opponent!)
- States: all possible positions on the board
- Actions: legal moves
- Reward: 1 or 0 (win/lose)

Open AI Five (OpenAI, 2019)(OpenAI et al., 2019)

- Environment: Dota 2
- States: encoding of the visible game state
- Actions: discrete actions

Rewards of OpenAI Five (OpenAI et al., 2019)

Name	Reward	Heroes	Description
Win	5	Team	
Hero Death	-1	Solo	
Courier Death	-2	Team	
XP Gained	0.002	Solo	
Gold Gained	0.006	Solo	For each unit of gold gained. Reward is not lost when the gold is spent or lost.
Gold Spent	0.0006	Solo	Per unit of gold spent on items without using courier.
Health Changed	2	Solo	Measured as a fraction of hero's max health.‡
Mana Changed	0.75	Solo	Measured as a fraction of hero's max mana.
Killed Hero	-0.6	Solo	For killing an enemy hero. The gold and experience reward is very high, so this reduces the total reward for killing enemies.
Last Hit	-0.16	Solo	The gold and experience reward is very high, so this reduces the total reward for last hit to ~ 0.4.
Deny	0.15	Solo	
Gained Aegis	5	Team	
Ancient HP Change	5	Team	Measured as a fraction of ancient's max health.
Megas Unlocked	4	Team	
T1 Tower*	2.25	Team	
T2 Tower*	3	Team	
T3 Tower*	4.5	Team	
T4 Tower*	2.25	Team	
Shrine*	2.25	Team	
Barracks*	6	Team	
Lane Assign†	-0.15	Solo	Per second in wrong lane.

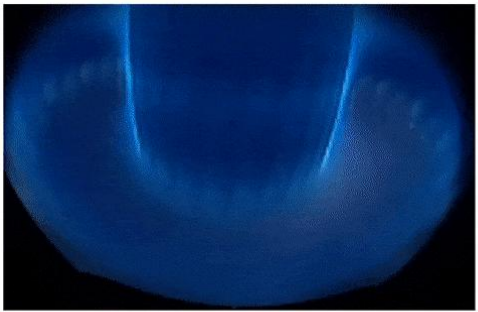
* For buildings, two-thirds of the reward is earned linearly as the building loses health, and one-third is earned as a lump sum when it dies.

† See item O.2.

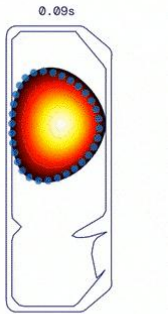
‡ Hero's health is quartically interpolated between 0 (dead) and 1 (full health); health at fraction x of full health is worth $\left(x + 1 - (1-x)^4\right)/2$. This function was not tuned; it was set once and then untouched for the duration of the project.

Stabilizing Nuclear Fusion (DeepMind, 2022)(Degrave et al., 2022)

- Environment: Tokamak fusion reactor
- States: measurments of the plasma inside the reactor
- Actions: controlling the voltages of the coils



View from inside the tokamak



Plasma state reconstruction

Rewards of Nuclear Fusion Stabilization

Extended Data Table 3 | Rewards used in the experiments

	Fundamental Capability	Elongated shape	ITER-like shape	Negative Triangularity	Snowflake	Droplets
Figure	Fig. 2	Fig. 3a, Extended Data Fig. 2a	Fig. 3b, Extended Data Fig. 2b, 3	Fig. 3c, Extended Data Fig. 2c	Fig. 3d, Extended Data Fig. 2d, 4c	Fig. 4
Shot	TCV#70915	TCV#70920	TCV#70600	TCV#70457	TCV#70755	TCV#69545
Reward Component	Transforms, Combiners (if necessary), and Weight (default=1)					
Diverted			Equal()	Equal()		
E/F Currents		SoftPlus(good=100, bad=50) GeometricMean()	SoftPlus(good=100, bad=50) GeometricMean()	SoftPlus(good=100, bad=50) GeometricMean()	SoftPlus(good=100, bad=50) GeometricMean()	
Elongation		SoftPlus(good=0.005, bad=0.2)		SoftPlus(good=0, bad=0.5)		
LCFS Distance	SoftPlus(good=0.005, bad=0.05) SmoothMax(-1)	SoftPlus(good=0.003, bad=0.03) SmoothMax(-1) weight=3	SoftPlus(good=0.005, bad=0.05) SmoothMax(-1) weight=3	SoftPlus(good=0.005, bad=0.05) SmoothMax(-1) weight=3	SoftPlus(good=0.005, bad=0.05) SmoothMax(-1) weight=3	
Legs Normalized Flux			Sigmoid(good=0.1, bad=0.3) SmoothMax(-5) weight=2			
Limit Point	Sigmoid(good=0.1, bad=0.2)	Sigmoid(good=0.2, bad=0.3)			Sigmoid(good=0.1, bad=0.2)	
OH Current Diff	SoftPlus(good=50, bad=1050)	ClippedLinear(good=50, bad=1050)	ClippedLinear(good=50, bad=1050)	ClippedLinear(good=50, bad=1050)	ClippedLinear(good=50, bad=1050)	ClippedLinear(good=50, bad=1050)
Plasma Current	SoftPlus(good=500, bad=20000)	SoftPlus(good=500, bad=30000)	SoftPlus(good=500, bad=20000) weight=2	SoftPlus(good=500, bad=20000) weight=2	SoftPlus(good=500, bad=20000) weight=2	Sigmoid(good=2000, bad=20000) weight=[1, 1]
R						Sigmoid(good=0.02, bad=0.5) weight=[1, 1]
Radius		SoftPlus(good=0.002, bad=0.02)		SoftPlus(good=0, bad=0.04)		
Triangularity		SoftPlus(good=0.005, bad=0.2)		SoftPlus(good=0, bad=0.5)		
Voltage Out of Bounds		Mean() SoftPlus(good=0, bad=1)	Mean() SoftPlus(good=0, bad=1)	Mean() SoftPlus(good=0, bad=1)	Mean() SoftPlus(good=0, bad=1)	
X-point Count		Equal()				
X-point Distance	Sigmoid(good=0.01, bad=0.15)		Sigmoid(good=0.01, bad=0.15) weight=0.5	Sigmoid(good=0.02, bad=0.15) weight=[0.5, 0.5]	Sigmoid(good=0.01, bad=0.15) weight=[0.5, 0.5]	
X-point Far	Sigmoid(good=0.3, bad=0.1) SmoothMax(-5)					
X-point Flux Gradient	SoftPlus(good=0, bad=3) weight=0.5		SoftPlus(good=0, bad=3) weight=0.5	SoftPlus(good=0, bad=3) weight=[0.5, 0.5]	SoftPlus(good=0, bad=3) weight=[0.5, 0.5]	
X-point Normalized Flux	SoftPlus(good=0, bad=0.08)		SoftPlus(good=0, bad=0.08)	SoftPlus(good=0, bad=0.08) weight=[1, 1]	SoftPlus(good=0, bad=0.08) weight=[1, 1]	
Z						Sigmoid(good=0.02, bad=0.2) weight=[1, 1]
Final Combiner	SmoothMax(-0.5)	SmoothMax(-5)	SmoothMax(-5)	SmoothMax(-0.5)	SmoothMax(-5)	GeometricMean()

Empty cells are not used in that reward. Any cell that does not specify a weight has an implicit weight of 1. Vector-valued weights (for example, Droplets: R) return several values to the final combiner. See Exended Data Table 4 for the descriptions of the different reward components and Extended Data Table 5 for the transforms, combiners and terminations. All of the terminations criteria were used for these experiments. Code for these rewards is available in the supplementary material.

Extended Data Table 4 | Reward components

Reward Component	Description
Diverted	Whether the plasma is limited by the wall or diverted through an X-point.
E/F Currents	The currents in the E and F coils, in amperes.
Elongation	The elongation of the plasma, this is its height divided by its width.
LCFS Distance	The distance in meters from the target points to the nearest point on the last closed flux surface (LCFS).
Legs Normalized Flux	The difference in normalized flux from the flux at the LCFS at target leg points.
Limit Point	The distance in meters from the actual limit point (wall or X-point) and target limit point.
OH Current Diff	The difference in amperes between the two OH coils.
Plasma Current	The plasma current in amperes.
R	The radial position of the plasma axis/centre, in meters.
Radius	Half of the width of the plasma, in meters.
Triangularity	The upper triangularity is defined as the radial position of the highest point relative to the median radial position. The overall triangularity is the mean of the upper and lower triangularity.
Voltage Out of Bounds	Penalty for going outside of the voltage limits.
X-point Count	Return the number of actual and requested X-points within the vessel.
X-point Distance	Returns the distance in meters from actual X-points to target X-points. Only X-points within 20cm are considered.
X-point Far	For any X-point that isn't requested, return the distance in meters from the X-point to the LCFS. This helps avoid extra X-points that may attract the plasma and lead to instabilities.
X-point Flux Gradient	The gradient of the flux at the target location with a target of 0 gradient. This encourages an X-point to form at the target location, but isn't very precise on the exact location.
X-point Normalized Flux	The difference in normalized flux from the flux at the LCFS at target X-points. This encourages the X-point to be on the last closed flux surface, and therefore for the plasma to be diverted.
Z	The vertical position of the plasma axis/centre, in meters.

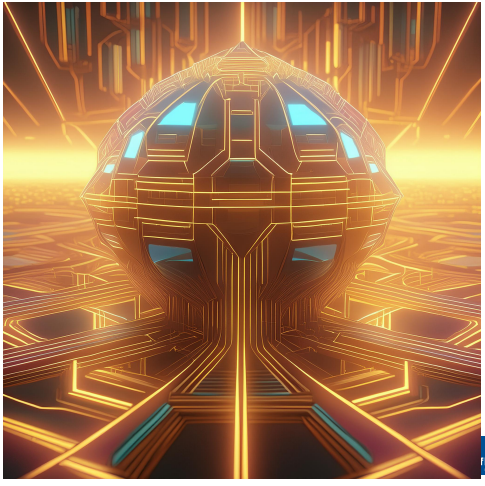
Description of reward components. All of these return an actual and a target value, and many allow time-varying target values. See Extended Data Table 3 for where and how they are used.

Autonomous Driving

- States: ?
- Actions: ?
- Rewards: ?

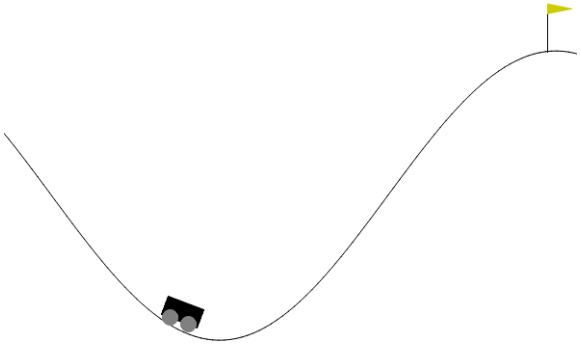
Outlook

- All these rewards are hard to design and tune.
- We will now look deeper at the fundamentals of reward design and then how to automatically generate rewards from intuitive human knowledge and skills!



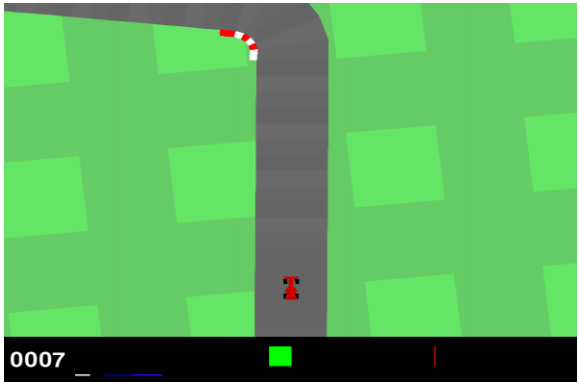
MountainCar

A car has to drive up a hill it is not strong enough to drive up.



CarRacing

A car drives along a randomly generated race track.



Fundamentals of Reward Design

Reward Design

- Reward the goal, not what you think is good!
- Iterate your reward, changing small things!
- When testing the reward, let the algorithm run long enough!

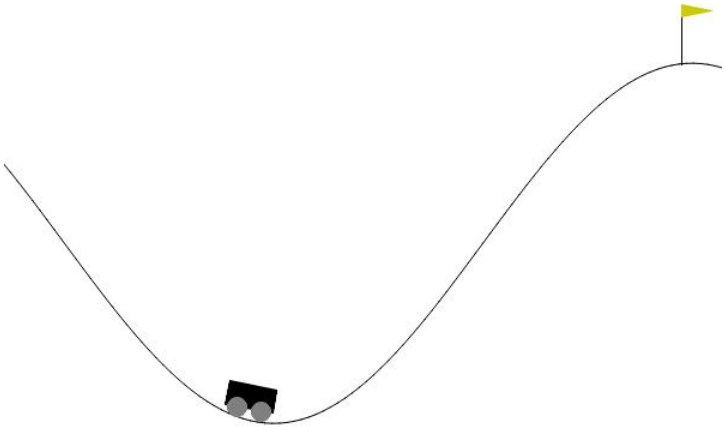
Reward Shaping

- If $r(s, a, s')$ is the reward of some environment, and $f(s, a, s')$ is some other function, then

$$r_f(s, a, s') := r(s, a, s') + f(s, a, s')$$

  is called **shaping reward**.
- $f(s, a, s')$ usually encodes **prior knowledge** about the environment.

Example: MountainCar (Towers et al., 2023)



Problems

- Requires a good understanding of what is useful for a task.
- Components of shaping rewards usually still very hard to tune!
- Only possible when features of state space are interpretable.

Imitation Learning

Problem Setting

We have access to an expert policy π_E, which is assumed to perform the given task perfectly. (Usually in the form of a dataset collected from the expert or a limited number of queries to the expert.)
We then want to create a policy π (e.g. a machine learning model) which performs the task similar to the expert.

Behavioral Cloning

- The simpleste method:
  - Gather State-Action (S, A) pairs from the expert
  - Fit a supervised model $\pi : \mathcal{S} \rightarrow \mathcal{A}$ to choose the expert action A in state S.

Inverse Reinforcement Learning

- Idea: Given a policy π, generate a reward $r(s, a, s')$ such that π is an optimal policy with respect to r.
- This reward r can then be used to train a RL agent, resulting in better generalization.
- In equations:

$$\underbrace{\max_{r \in \mathcal{R}} \underbrace{\left(\min_{\pi \in \Pi} -\mathcal{H}^{\gamma}(\pi) - \mathbb{E}_{\pi}[r(s,a)] \right)}_{\text{maximize imitator performance}} + \mathbb{E}_{\pi_E}[r(s,a)]}_{\text{distinguish between imitator and expert}}$$

Regularizing

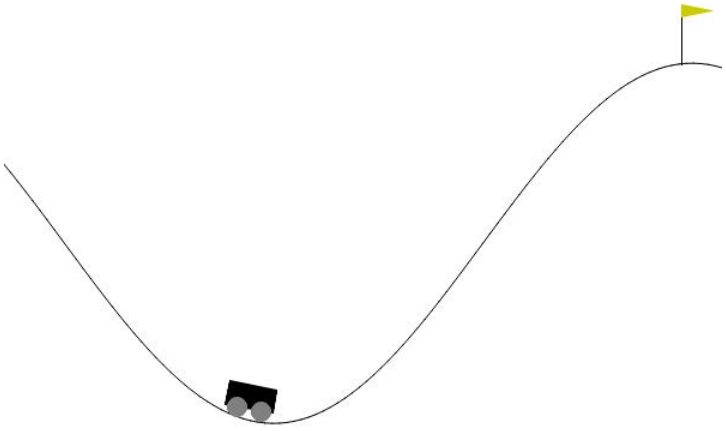
Add regularizing function:

$$\max_{r \in \mathcal{R}} \left(\min_{\pi \in \Pi} -\mathcal{H}^{\gamma}(\pi) - \mathbb{E}_{\pi}[r(s,a)] \right) + \mathbb{E}_{\pi_E}[r(s,a)] - \psi(r),$$

Problem: Back and forth between r and π!

(Garg et al., 2021) and (Al-Hafez et al., 2023) combine both objectives into a single one!

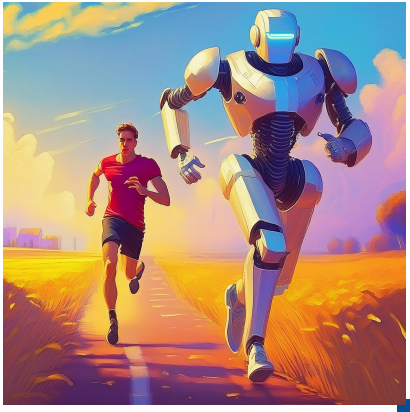
Example: MountainCar (Towers et al., 2023)



Improving the Policy

Improving the Policy

Now that we have found rewards for training initial policies/imitated an expert, we want to train agents surpassing human performance.



Regularize Reward towards Imitated Policy

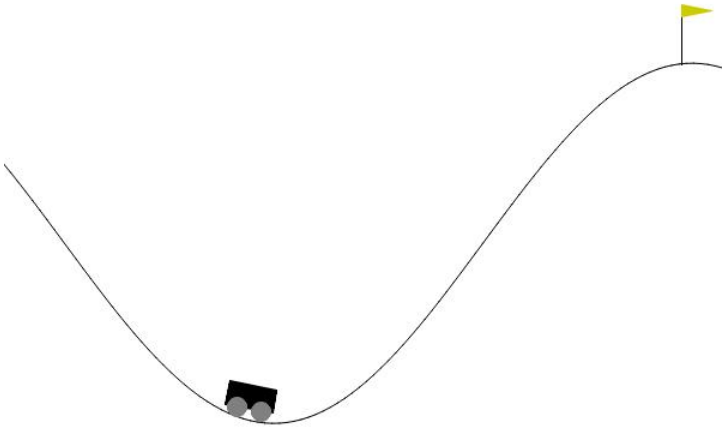
Similar to shaping, we transform the reward by

$$\tilde{r}(s, a, s') = r(s, a, s') + \alpha \, D_{\mathrm{KL}} \left(\pi \| \pi_I \right),$$

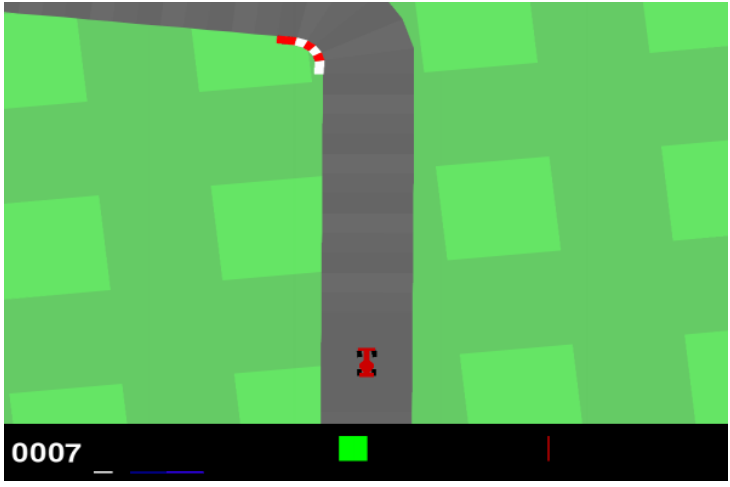
where π_I is the imitated policy frozen after completing imitation learning and $D_{\mathrm{KL}}(\cdot\|\cdot)$ is the Kullback-Leibler divergence.

(The algorithm we will use does this in a slightly more sophisticated way, similar to Soft Actor-Critic, (Haarnoja et al., 2018))

Example: MountainCar (Towers et al., 2023)



Time for the Race Track!



References

- Al-Hafez, F., Tateo, D., Arenz, O., Zhao, G., & Peters, J. (2023). **Ls-iq: Implicit reward regularization for inverse reinforcement learning.** *Eleventh International Conference on Learning Representations (ICLR).* https://openreview.net/pdf?id=o3Q4m8jg4BR (cit. on p. 24).
- DeepMind. (2018). ***Alphazero.*** Retrieved May 3, 2024, from https://www.deepmind.com/blog/alphazero-shedding-new-light-on-chess-shogi-and-go (cit. on p. 6).

DeepMind. (2022). ***Accelerating fusion science through learned plasma control.*** Retrieved May 3, 2024, from https://www.deepmind.com/blog/accelerating-fusion-science-through-learned-plasma-control (cit. on p. 9).

Degrave, J., Felici, F., Buchli, J., Neunert, M., Tracey, B., Carpanese, F., Ewalds, T., Hafner, R., Abdolmaleki, A., de las Casas, D., Donner, C., Fritz, L., Galperti, C., Huber, A., Keeling, J., Tsimpoukelli, M., Kay, J., Merle, A., Moret, J.-M., … Riedmiller, M. (2022).**Magnetic control of tokamak plasmas through deep reinforcement learning.** *Nature*, *602*(7897), 414–419. https://doi.org/10.1038/s41586-021-04301-9 (cit. on p. 9).

Garg, D., Chakraborty, S., Cundy, C., Song, J., & Ermon, S. (2021). **Iq-learn: Inverse soft-q learning for imitation.** *CoRR, abs/2106.12142*. https://arxiv.org/abs/2106.12142 (cit. on p. 24).

Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018, October). **Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor.** In J. Dy & A. Krause (Eds.), *Proceedings of the 35th international conference on machine learning* (pp. 1861–1870, Vol. 80). PMLR. https://proceedings.mlr.press/v80/haarnoja18b.html (cit. on p. 28).

OpenAI. (2019). ***Openai five.*** Retrieved May 3, 2024, from https://openai.com/five/ (cit. on p. 7).

OpenAI, : Berner, C., Brockman, G., Chan, B., Cheung, V., Dębiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., Józefowicz, R., Gray, S., Olsson, C., Pachocki, J., Petrov, M., Pinto, H. P. d. O., Raiman, J., ... Zhang, S. (2019). **Dota 2 with large scale deep reinforcement learning.** https://doi.org/10.48550/ARXIV.1912.06680 (cit. on pp. 7, 8).

Sutton, R. S., & Barto, A. G. (2018). ***Reinforcement learning: An introduction.*** MIT press. (Cit. on p. 4).

Towers, M., Terry, J. K., Kwiatkowski, A., Balis, J. U., Cola, G. d., Deleu, T., Goulão, M., Kallinteris, A., KG, A., Krimmel, M., Perez-Vicente, R., Pierré, A., Schulhoff, S., Tai, J. J., Shen, A. T. J., & Younis, O. G. (2023, March). **Gymnasium.** https://doi.org/10.5281/zenodo.8127026 (cit. on pp. 18, 25, 29).