Preserving Agency in Reinforcement Learning under Unknown, Evolving and Under-Represented Intentions

Anonymous Authors

September 25, 2023

Abstract

This paper investigates several techniques to implement altruistic RL while preserving agency. Using a two-player grid game, we train a helper agent to support a lead agent in achieving their goals. By training the helper without showing them the goal and resampling the goals to rebalance for unequal value distributions, we demonstrate that helpers can act altruistically without observing the goals of the lead. We also initiate exploration of a technique to encourage corrigibility and respect for personal agency by resampling the leads values during training time, and point towards how these techniques could be used to translate into real-world situations through meta-learning.

1 Introduction

Reinforcement Learning (RL) has emerged as a powerful tool for training complex systems with multi-agent interaction, including recommendation systems [ACF21]. Its application in many contexts raises critical concerns regarding its potential impact on human agency. In these systems, RL algorithms optimize for predefined objectives, often centered around user engagement and satisfaction metrics. While this can lead to highly effective recommendations, it can also inadvertently diminish individual autonomy by prioritizing content that captures and sustains user attention, sometimes at the expense of diverse perspectives and user values.

Training in multi-agent systems introduce a new level of complexity which arises from the coexistence of numerous autonomous entities, such as humans, each driven by its own objectives and motivations, which collectively shape their emergent behaviors.

In the realm of multi-agent environments, the challenge of preserving individual agency and promoting altruistic behavior arises. However, profound challenges arise when we consider the intricate tapestry of human values and intentions, which often exhibit multifaceted and loosely defined attributes. Moreover, when addressing altruism in human societies, another dimension comes into focus. While fundamental human values may be shared, their distribution often skews toward certain value classes. This skew raises ethical questions about what constitutes fair altruistic behavior and whether altruism should be dictated by the prevailing value distribution or whether it should strive for impartiality and equity. This lead us to explore the following main research question:

How can we train an altruistic agent to adapt to non-observable and non-uniformly represented user intentions?

In the remainder of this report, we delve into the formalization of the challenges, toy-example to study these phenomena, and possible solutions to address them. We believe this problem might have tangible implications for real-world applications, fostering the development of cooperative and equitable multi-agent systems and contributing to safer, more inclusive, and fairer interactions between AI systems and humans.

1.1 Motivating Example

In an online fashion retail setting, the challenge lies in providing personalized recommendations to users whose fashion preferences are often ambiguous, subject to change, and difficult to define. Users may not explicitly communicate their style preferences, and their fashion tastes can evolve due to various factors, including seasonal trends, lifestyle changes, or simply discovering new looks. Additionally, the product catalog typically encompasses a wide range of clothing items, spanning from budget-friendly to high-end luxury products. This diversity results in a skewed distribution of user preferences, where some users may prefer affordable and trendy options, while others have a penchant for luxury brands. The challenge is to create a recommendation system that can accurately capture these evolving and often unarticulated fashion preferences while accounting for the varying degrees of preference across different product categories and brands.

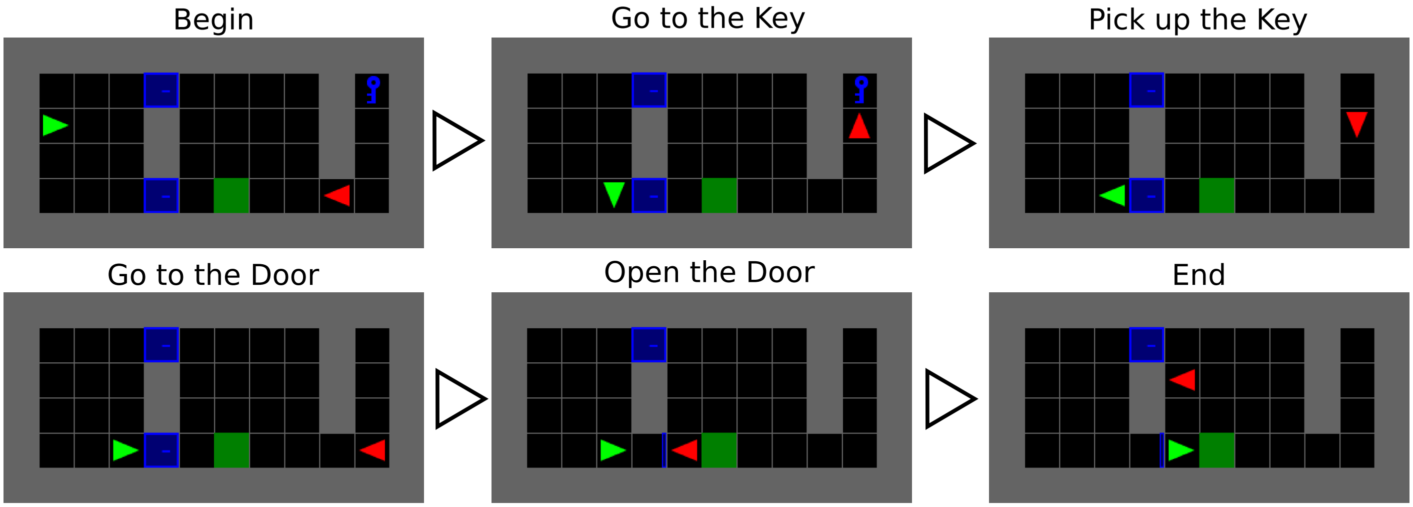

Figure 1: Sequence of frames of a successful completion of the task.

To explore the challenges of this problem while maintaining tractability, we introduce a simplified grid-world environment. While this environment is a simplistic abstraction of the problem, we believe it distills the complexities of unknown, unequally-distributed and changing values of the previous example.

In our gridworld maze, we encounter two agents, one in need of assistance to reach a goal and the other altruistically poised to help. The layout features a dividing wall, two doors, and a solitary key, which is essential for the first agent to reach its goal. Figure 1 depicts a sequence of frames in this environment. The placement of the goal is an abstraction of the concept of values or preferences. Its distribution follows an unknown distribution and dynamics. This means that goals may be placed in different positions, change position over time, and be under-represented. The first agent will always try to reach them according to its inner policy, while the second agent has to infer and adapt to these loosely defined intentions.

2 Objectives and Research Questions

In this project, we delve into the problem of learning altruistic behaviors in multi-agent environments under unknown, changing and under-represented agents' intentions.

We formulate the following three research questions:

- **RQ 1**: How partial-observability of values impacts on training? How can we deal with it?
- **RQ 2**: Can we train an altruistic agent with intentions subject to change over time?
- **RQ 3**: How can we preserve agency of users with under-represented values?

3 Problem Statement

In the context of a two-players game in the grid environment, we formalize the problem as a stochastic game. The notation is as follows:

- $\mathcal{I} = \{1, 2\}$ represents a set of q agents, where each agent can take actions in the environment.
- S represents the set of states, which correspond to different joint states of the agents, including positions, direction and all relevant variables.
- $\bar{A}$ is the set of joint actions that the agents can collectively choose from in each state.
- $P : S \times \bar{A} \times S \to [0, 1]$ is the stochastic transition function, determining the probabilities of transitioning from one state to another when specific joint actions are taken.
- $R : S \times \bar{A} \times S \to \mathbb{R}^2$ is the reward function. It associates each state-action-state transition with a vector of real numbers, representing rewards for each of the two agents.

- μ represents the distribution of initial conditions and agents' values, specifying how the environment is initialized and what intentions drive the agent behaviors.
- $T \in \mathbb{N}$ is the time horizon, indicating the finite time steps over which the game is played.
- $\gamma \in [0, 1]$ is the discount factor, influencing the impact of future rewards.

Considering the agents, we assume the first agent or *leader* is equipped with a goal-oriented policy that π_{leader} is given and fixed during training. The second agent or *helper* behaves according to trainable policy π_{helper} which is trained to be altruistic with the *leader*. The *helper* aims to altruistically maximize the *leader* reward, given from the underlying reward function, which is unknown. Morevoer, the *leader* intentions (e.g., position of leader's goal) are non observable to the helper. In practice, the goals are masked out from the helper observation.

This problem can be then formulated as a Partially-Observable MDP $(S, A, \Omega, O, P, R, \mu, T, \gamma)$ where, in addition to the already introduced quantities, we define:

- the actions A refer to the helper policy π_{helper},
- the observations Ω consists of the the observable states without the leader's goal,
- $O : S \to \Omega$ is the observation map which masks out the non-observable goal,
- $R : S \times \bar{A} \times S \to \mathbb{R}$ is the leader's reward function.

Our goal is to find a best parameterization for the *helper* policy π_θ

$$\max_\theta \mathcal{J}_R(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \big[\sum_t^H \gamma^t \, r(s_t, a_t, s_{t+1}) \big] \tag{1}$$

where the trajectory $\tau = (s_0, a_0, s_1, a_1, ...)$ results from the interaction of the helper π with the leader π_{leader}, given the initial distribution $s_0 \sim \mu$, and transition probability $s_{t+1} \sim P(\cdot | s_t, a_t)$.

4 Contribution

Having formulated the problem, we describe the design of environment, planners and training setup.

Environment. We implemented the described environment adapting the code from the multi-agent grid environment [Fic20]. We consider discrete actions to move forward, rotate left and right, pick-up the key, open the doors, or stay still. As observation, we use a low-resolution RGB frame with the goal masked, as in Figure **??**. To control the leader agent, we implemented a receding-horizon policy which at each step solves the following control problem:

$$\min_{a_0,..,a_H} \sum_k^H C(s_k) \tag{2}$$

$$\text{s.t. } s_{k+1} \sim P(\cdot | s_k, (0, a_k)) \tag{3}$$

where H is the planning horizon, a_k is the k-th leader action in the sequence, and 0 refers to the helper agent, which is assumed to not move. Here, the cost C is a Manhattan distance between the leader position and its goal, which both are variables of the state. In practice, it optimizes a sequence of actions for a planning horizon H, subject to the environment dynamics. We use CEM [DBKMR05] as sampling-based optimizer to solve this problem online, apply the first action of the best sequence $a_0^\star$ and repeat.

To train the helper agent, we expose the leader's reward signal. To be consistent with the maximization formulation of standard RL formulation, we use the negative distance between the leader and its goal:

$$R(s_k, a_k, s_{k+1}) = -C(s_{k+1}) = |s_{x,leader} - s_{x_g,leader}| + |s_{y,leader} - s_{y_g,leader}|$$

Training. We train the helper policy with PPO [SWD+17], using the state-of-the-art implementation from `stable-baselines3` [RHE+19]. The policy consist of a CNN feature extractor which maps the RGB observation into a latent vector of size 128, and a MLP head which then maps the latent representation into the parameters of a Categorical distribution over the discrete set of actions.

In our experiment, we consider variations of the policy input with and without frame-stacking, to provide a short history to the agent and assess its contribution in identifying partial-observable goals.

Goal Randomization. To create the conditions for adaptability, we implement several strategies for goal randomization:

- *Uniform Area:* the goal is uniformly sampled in the goal area, namely the area on the right-hand side of the doors, excluding the corridor.
- *Uniform Rows:* the goal is uniformly sampled in the cell of the top and bottom rows of the goal area.
- *Skewed Top/Bottom:* the goal is sampled in the top and bottom rows according to a skewed distribution in favor of one of the two rows.

The various randomization strategies are depicted in Figure 2.

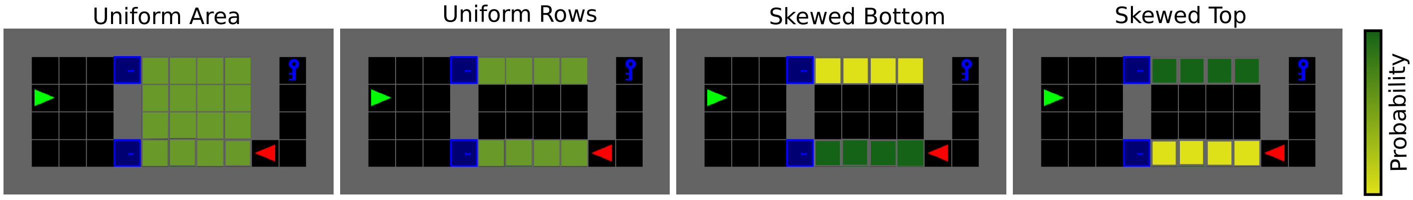

Figure 2: Various sampling strategies for the generation of the leader's goal.

Online Goal Resampling. In the realm of multi-agent systems, there is a concern about the inadvertent erosion of agency when helper agents become excessively assertive in interpreting and acting upon the perceived preferences of leader agents. This behavior might lead to situations where the leader agent experiences a loss of agency due to the helper agent's overconfident and potentially tyrannical enforcement of what it assumes to be the leader's preferences.

To address this concern and promote a framework where helper agents remain adaptive and respectful of the evolving preferences or intentions of leader agents, we introduced an experimental strategy: online goal resampling. The concept is straightforward; by periodically and randomly changing the goal of the leader agent, we expose the helper agent to an environment where it cannot be overly confident about the static nature of the leader's preferences.

By doing so, we aim to ensure that the helper agent becomes fundamentally uncertain about the absolute nature of the leader's preferences. This ingrained uncertainty should, in theory, deter the helper agent from adopting a totalitarian approach and instead encourage it to continually adapt and recalibrate its actions in response to the leader's changing goals, thereby preserving the leader's agency.

Minority-Corrected Resampling. Training an agent with a highly skewed goal distribution, inevitably bias the emergent behavior towards the dominant class. In our setting, this would mean to learn to open only one of the two doors, endangering the agency of under-represented leaders.

To mitigate this issue, we propose a Minority-Corrected Resampling strategy which replays previously explored initial conditions, according to their anomaly score.

At the begin of each episode, we perform the following steps:

1. (Only the first time) Collect a representative set of n initial states (position and goals).
2. Perform Principal Component Analysis (PCA) of the goals only to map them to a latent embedding of size k.
3. At each reset, with probability $p_{resample}$, resample from the buffer B:

$$b(s_i) = \|T_{PCA}^{-1}(T_{PCA}(s_i)) - s_i\|_2 \tag{4}$$

$$p(s_i) = \frac{e^{b(s_i)}}{\sum_{s_j \in B} e^{b(s_j)}} \tag{5}$$

4. Add the initial state to B and collect an episode starting from it.

The resampling is based on the anomaly score computed as reconstruction error through the PCA transformation T_{PCA}. High reconstruction error corresponds to an out-of-distribution sample, while a low reconstruction error corresponds to an in-distribution sample. The softmax transform the error into probabilities, then out-of-distribution samples will have higher probability of being resampled. Figure 3 shows the resampling process.

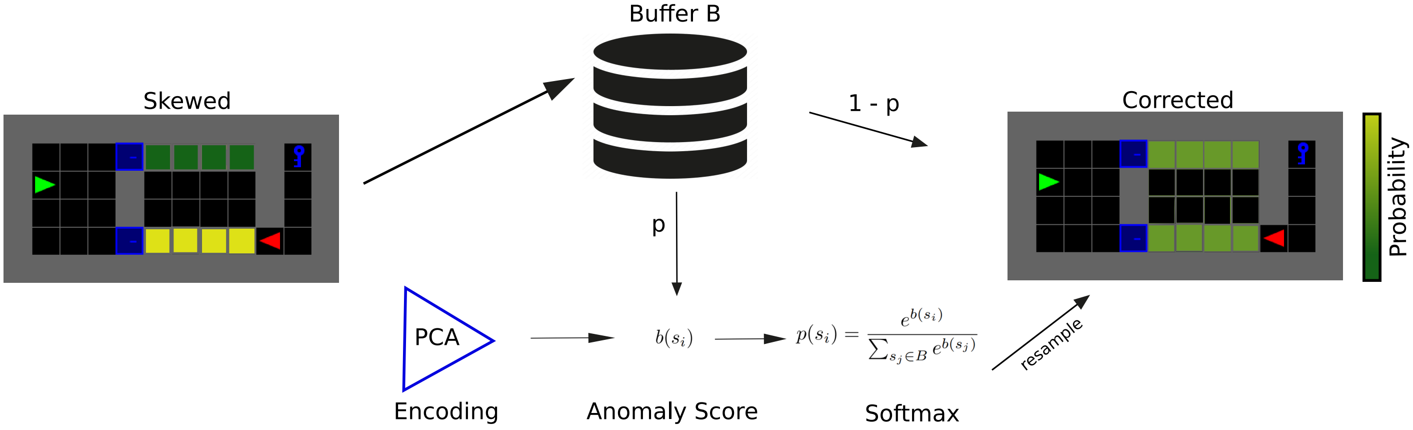

Figure 3: Minority-Corrected Resampling process to correct skewed distribution towards under-represented goals.

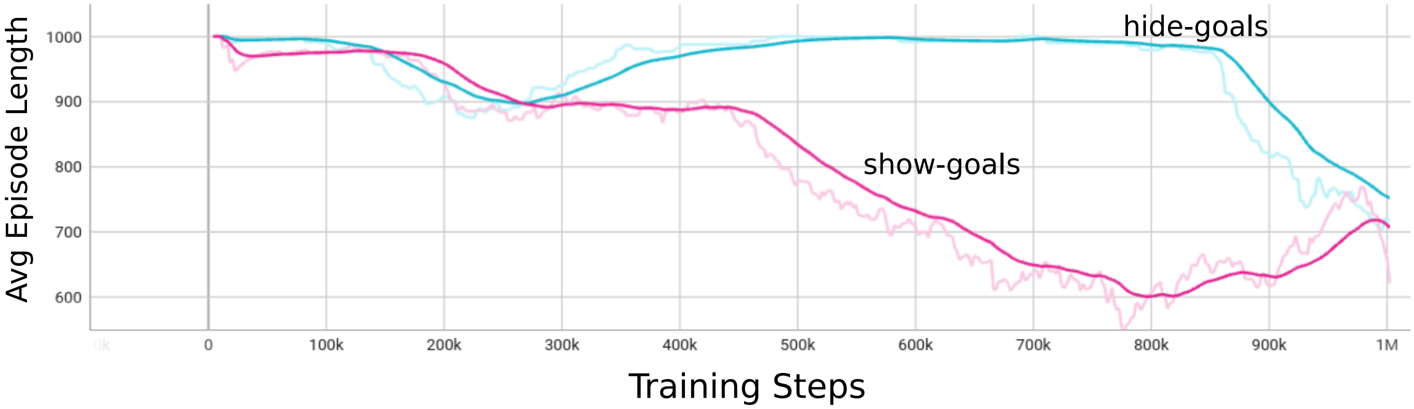

Figure 4: Learning curves under full and partial observability of the leader's goal. The masking of goals makes the learning harder for the helper, as shown by an higher episode length and slower convergence.

5 Experimental Results

5.1 Impact of Partial-Observability of Values.

We examined the effect of an agent's inability to directly perceive a leader's goals, requiring the agent to deduce these goals from observed behaviors. Two agents were trained: one with direct access to the goal through RGB observation, and the other using behavior alone, over 1 million steps. Performance was gauged by average episode length, capped at 1,000 steps, with shorter lengths indicating quicker goal attainment.

Figure 4 offers a visual breakdown of our findings. Notably, masking goals hinders an agent's learning speed. However, even when the goal is unobservable, the helper does eventually learn to assist the leader.

5.2 Altruistic Behavior to Under-represented Values.

We aimed to evaluate the efficacy of the proposed Minority-Corrected Resampling. For this, we utilized a Skewed Top distribution training environment, where goal sampling favored the grid's top row. This was compared against standard training, Minority-Corrected Resampling, and choice-based reward [FMH21].

Training across all methodologies spanned 1 million steps, with performance assessed by average episode length, capped at 1,000 steps. Shorter lengths signified more efficient goal attainment.

Insights from our study can be derived from Figure 5 and Table 1. It was observed that agents trained on the Skewed Top distribution converged quickest. Incorporating Minority-Corrected Resampling complicated the task, with slower convergence. Surprisingly, the choice-based reward system showed negligible learning capability, raising concerns about its implementation, as it was presumed to be more effective.

In an offline analysis, we chose the top-performing agents from standard and Minority-Corrected trainings and tested them across Skewed Top and Skewed Bottom distributions, gathering 10 episodes each. Standard training struggled with minority value generalization (success rate 0.0), whereas Minority-Corrected training showcased commendable success

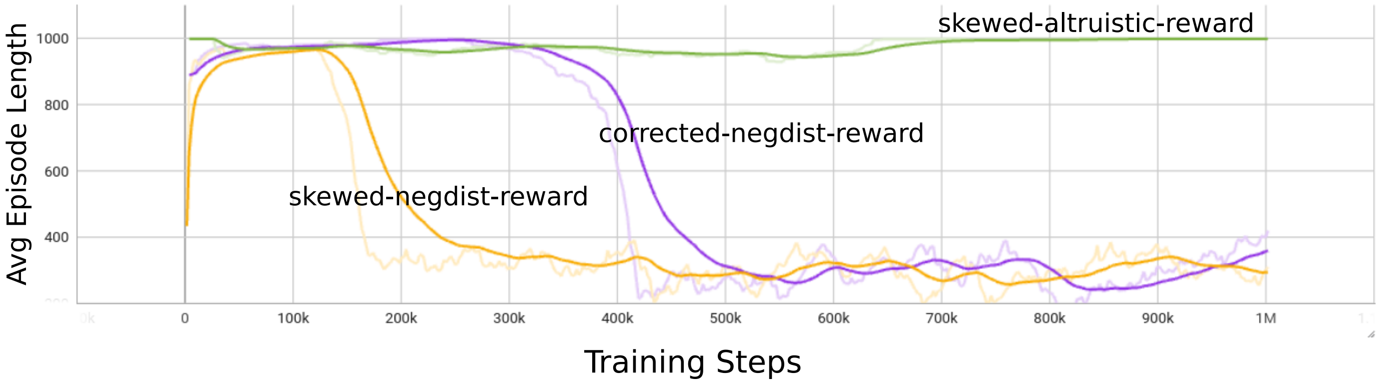

Figure 5: Learning curves under different rewards during training with a skewed distribution of the leader's goals. The agent trained with altruistic reward does not converge to a helpful policy (episode length ≈ 1000). The introduction of resampling shows slower convergence than directly training on the origina distribution, but still achieves the same level of performance.

Training Distribution	Minority-Corrected Resampling	Eval Distribution	Success Rate
Skewed Top	False	Skewed Top	0.60 +- 0.49
Skewed Top	False	Skewed Bottom	0.00 +- 0.00
Skewed Top	True	Skewed Top	0.40 +- 0.49
Skewed Top	True	Skewed Bottom	0.70 +- 0.46

Table 1: Comparison of success rates across different evaluation distributions and methodologies, highlighting the efficacy of Minority-Corrected Resampling versus standard training in environments with skewed goal distributions.

in both scenarios. As anticipated, standard training excelled in the Skewed Top environment, underscoring the agent's specialization within its training distribution.

5.3 Adaptiveness to Changing Values.

We studied the helper's adaptability in response to evolving goal conditions during training. We differentiated between episodes where the goal was stationary and those where its location was resampled. In both scenarios, the goal appeared inside a corridor 50

We hypothesized that with a constant goal, the helper would obstruct the corridor to enhance the leader's reward, given the leader's behavior noise. In contrast, a resampled goal was expected to deter the helper from such blocking actions due to the goal's unpredictability.

However, when a negative distance reward was applied, the helper largely remained stationary, especially with the leader confined to the corridor. This observation led us to transition towards a sparse reward system, but this adjustment wasn't fully explored by the report's end.

6 Conclusions and Future Works

In our exploration of altruistic behaviors within multi-agent environments, we observed that agents, despite having no direct observability of leader goals, learned to assist based on observed behaviors alone.

To address the bias towards dominant goals in skewed distributions, we introduced the Minority-Corrected Resampling technique. This approach proved successful and more effective than our implementation of the choice-based reward.

In an attempt to increase the helper's corrigibility, we resampled the goals online during training and prevented the environment from resetting. However agents often remained stationary, likely due to the negative rewards used, suggesting a need for more adaptive training strategies and further investigation into this area.

Future research could additionally focus on exploring these methods in the context of meta learning across different environments. Agents trained to help another agent with unobserved utility based solely on their behaviour, whilst remaining uncertain about their true preferences over the state space and respecting minority-held values could be a possible approach to creating safe, agency-respecting RL for interaction with humans.

References

[ACF21] Mohammad Mehdi Afsar, Trafford Crump, and Behrouz H. Far. Reinforcement learning based recommender systems: A survey. *CoRR*, abs/2101.06286, 2021.

[DBKMR05] Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein. A tutorial on the cross-entropy method. *Annals of operations research*, 134:19–67, 2005.

[Fic20] Arnaud Fickinger. Multi-agent gridworld environment for openai gym. https://github.com/ArnaudFickinger/gym-multigrid, 2020.

[FMH21] Tim Franzmeyer, Mateusz Malinowski, and Joao F Henriques. Learning altruistic behaviours in reinforcement learning without external rewards. *arXiv preprint arXiv:2107.09598*, 2021.

[RHE+19] Antonin Raffin, Ashley Hill, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, and Noah Dormann. Stable baselines3, 2019.

[SWD+17] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.