Model-Free Reinforcement Learning for Sensorless Adaptive Optics

Ioannis Koutalios (s3365530)[1] **Juri Morisse(s3737764)**[1]

Abstract

Light carries information across interstellar distances. Capturing such light with telescopes allows astronomers to investigate galaxies, planets, or black holes. However, atmospheric and other disturbances pose a challenge. To overcome these, adaptive optics (AO) systems (e.g. telescopes) consist of many deformable mirrors (DM) which mitigate noise when correctly adjusted. Recent work has focused on applying reinforcement learning (RL) techniques to automatically control the DM. However, prior work on RL AO control for astronomical imaging relied on sensors directly measuring disturbances' effects on images. In this work, we propose a sensorless approach in which a model-free Soft Actor-Critic agent learns to control DM to center and sharpen images. We find that our agent is successful in centering images and also in sharpening images. The performance in the latter task, however, strongly depends on the number of DM. Thus, our results demonstrate that the main challenge in RL-based AO control lies in its high action dimensionality.

1. Introduction

How do galaxies form and evolve? What are the characteristics of distant planets? How do black holes affect their surrounding? To find answers to these questions, astronomers rely on any information they can gather from sources at an interstellar distance. The most important carrier of such information is light captured through telescopes on Earth or in space. The obtained measurements can then be used to compute properties like velocity, composition, or temperature of interstellar objects or to simply visualize the unimaginable. However, recording light from sources billions of light years away is sensitive to even the smallest disturbances in the atmosphere or from other sources like vibrations. To counter those disturbances, adaptive optics (AO) systems consist of many individual deformable mirrors (DM) that can be adjusted to balance out disturbances and obtain a clean measurement.

Controlling a telescope's DM is a high-dimensional decision problem that follows a constant loop of adjusting mirrors and checking the adjustment's effect on the acquired image. Due to the sequential nature of this process and a clearly defined goal measure (reducing the noise), prior work increasingly focused on applying reinforcement learning (RL) to AO control (Durech et al., 2021; Landman et al., 2021; Nousiainen et al., 2021; Pou et al., 2022; Parvizi et al., 2023a;b). However, recent work on RL-based AO control for astronomical imaging relied on systems that contain a sensor measuring the disturbance present in images and trained RL agents to choose actions based on their effect on this sensor (Nousiainen et al., 2021; Landman et al., 2021).

In this work, we demonstrate that RL can also be used for sensor-free AO control in the context of astronomical imaging. We argue that this approach is advantageous as it reduces the overall system complexity by removing the sensor component and with it any potential noise or bias such a sensor can introduce. We show that a pre-implemented and only slightly modified soft actor-critic (SAC) agent can learn AO control based only on the observed image. We evaluate our approach on several tasks for centering and sharpening images. We find that our SAC agent successfully learns to center images. Sharpening images proves to be more challenging as it makes use of the full number of DM. Experimenting with reducing the action space through grouping DM revealed that our agent can control several DM successfully but that increasing the number of controllable DM causes both performance loss and longer training. These results confirm that the key challenge for RL-based AO control lies in the large action dimensionality.

This work aims to provide a new RL-based approach for AO control in the context of astronomical imaging. To this end, we make the following contributions:

- We show that a model-free SAC agent can perform sensorless AO control for centering and sharpening astronomical images.
- We demonstrate that the key challenge for applying RL to AO lies in the high action dimensionality required by realistic AO systems for astronomical imaging.
- Through the discussion of our work and the review of related work, we identify the need for a universal RL-based AO control benchmark that would allow comparability across related works.

In the following, we start by giving an overview of the underlying concepts of AO and RL, and the used SAC algorithm before providing a concise summary of related work. Following that, we give a detailed description of our experiments followed by a presentation of our results. Afterward, we critically discuss the shortcomings of our work and potential for future work in the field before closing with a conclusion.

2. Preliminaries

2.1. Adaptive Optics

AO is a technique used to correct for disturbances in optical systems. In astronomy, for example, it can be used to correct for the blurring of images caused by the atmosphere, through a set of sophisticated DM, allowing for sharper images to be taken.

Usually, an AO system consists of a wavefront sensor, a DM, and a control system. The wavefront sensor measures the wavefront distortion caused by the atmosphere and the control system calculates the required correction to be applied to the DM. The change of the DM then causes the correction to the wavefront, allowing for a sharper image to be taken (Roddier, 2004). This operation is repeated at a high frequency, allowing for real-time correction of the wavefront as in a closed feedback loop.

The phase variations of the wavefront can be described by a set of Zernike polynomials, which are a complete set of orthogonal functions on the unit circle (Beckers, 1993). The Zernike polynomials are used to describe the wavefront distortion in terms of a set of coefficients, which are then used to calculate the required correction to be applied to the DM. The number of Zernike modes used to describe the wavefront distortion is called the order of the system. The higher the order of the system, the more accurate the correction applied to the wavefront will be.

Guide stars are used to measure the wavefront distortion. A guide star is a bright star (usually < 15 mag) in the field of view of the telescope, which is used as a reference to measure the wavefront distortion. The sky coverage of an AO system is limited by the number of guide stars available in the field of view of the telescope at around 10% (Davies & Kasper, 2012). For this reason, we often use laser guide stars, which are artificial stars created by shining a laser into the atmosphere and can cover the entire sky.

2.2. Reinforcement Learning

Reinforcement Learning (RL) is a branch of machine learning concerned with training agents to make sequential decisions that maximize an observed reward within a certain environment. As such, RL is most commonly researched in games where possible actions, the environment, and the reward are usually clearly defined. However, RL is by no means limited to the virtual domain and can be applied to real-life problems whenever the problem can be formulated as a sequential decision-making problem for which the quality of decisions can be quantified by a reward function. Such real-life applications most commonly are concerned about automatic control, e.g. for self-driving vehicles (Kiran et al., 2022) or, like in this case, for controlling adaptive optics to improve astronomical imaging.

The specific type of problem for which RL is a suitable approach can be formally described as Markov Decision Processes (MDP). A MDP is formalized as a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R} \rangle$. Here:

- $\mathcal{S}$ is the set of possible states of the environment.
- $\mathcal{A}$ is the set of actions available to the agent.
- $\mathcal{P}$ is the transition function, where $\mathcal{P}(s'|s, a)$ denotes the probability of transitioning to state s' when performing action a in state s.
- $\mathcal{R}$ is the reward function, where $\mathcal{R}(s, a, s')$ defines the immediate reward associated with transitioning from state s to s' by taking action a.

The agent interacts with the environment in a sequence of discrete time steps, $t = 0, 1, 2, \ldots$, by selecting actions according to a policy π, which maps states to actions. The goal is to learn the optimal policy π^* that maximizes the expected cumulative reward over time.

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \cdot \mathcal{R}(s_t, a_t, s_{t+1})\right] \tag{1}$$

Here, $\gamma \in [0, 1)$ is the discount factor that controls the trade-off between immediate and future rewards.

In the context of our adaptive optics (AO) application, states represent the currently obtained image, actions correspond to adjustments of the deformable mirrors, the transition function is unknown, and the reward function depends on the specific task. For example, the reward for image sharpening is the Strehl ratio which measures image sharpness.

2.3. Soft Actor-Critic

Soft Actor-Critic (SAC) is a policy-based, model-free reinforcement learning algorithm designed for continuous action spaces first proposed by Haarnoja et al. (2018). Due to its characteristics, it is very suitable for our given AO problem as we are dealing with continuous actions and the unknown transition function P is no issue for model-free RL algorithms that do not attempt to estimate P anyway.

As an instance of actor-critic algorithms, SAC uses an actor for action selection and a critic for value estimation. Dividing these two tasks among two agents that are trained separately allows for improved stability and is a feature of many RL algorithms. SAC's distinctive feature is its incorporation of entropy regularization which encourages exploratory decision making.

In this work, we use the SAC implementation available through Stable Baselines. For further details on this specific implementation, we refer the reader to the respective documentation [1].

While we also experiment with an Advantage Actor-Critic (A2C) agent, a variant of the Asynchronous Advantage Actor Critic (A3C) (Mnih et al., 2016), we generally saw SAC outperforming the A2C agent. Thus, we mainly focused on experimenting with the SAC agent and decided to not include a dedicated section on the A2C agent.

3. Related Work

AO can be applied not only to astronomical imaging but to any imaging task where a DM can be controlled to mitigate disturbances affecting the image. Just like for its application, there is also a large variety of previously proposed methods for AO control. In this section, we will give a concise overview of such previous work, focusing on RL-based methods, and discuss how our work integrates into the field.

A common approach of previous work is to use RL for predicting the temporal evolution of disturbances from past experiences using a Wavefront Sensor (WFS). Landman et al. (2021) employ a model-free RL approach based on the Deterministic Policy Gradient algorithm to optimize a Recurrent Neural Network controller to mitigate the effect of telescope vibrations through tip-tilt control of a DM. Nousiainen et al. (2021) follow a model-based approach, where the model is learned via supervised learning from previous experiences and can be used to estimate the image distortions resulting from changes to the DM. Using this model, an RL agent is trained via Probabilistic Ensemble Trajectory Sampling (PETS) to find actions that maximize the expected future rewards. Model updating and agent training are repeated consecutively until convergence indicates a (local) optimum has been reached. Further work has explored the use of multi-agent reinforcement learning (Pou et al., 2022).

While differing in their algorithmic choices and their use of a dynamics model, both Landman et al. (2021) and Nousiainen et al. (2021) train an RL agent on input from a WFS. In contrast to that, we train the agent directly on images. While this training directly from images has been done for RL-based control of AO, it was either applied to microscopy (Durech et al., 2021), satellite-to-ground communication (Parvizi et al., 2023a;b), or a simulated laboratory imaging setup (Ke et al., 2019). To the best of our knowledge, no previous work has been done on RL-based AO control for astronomical imaging trained directly on the raw images.

4. Experiments

Our experiments are conducted using the set of Adaptive Optics environments provided by Rico Landman [2]. The original environments were modified slightly to fit our needs [3]. We then created a set of experiments to test the performance of the SAC and A2C algorithms in different environments. The experiments were conducted on multiple identical machines with an Intel Core i5-7500 CPU @ 3.40GHz, 8GB of RAM, and no GPU.

4.1. Environments

There are a total of 4 environments provided by Rico Landman, of which we used 3. The first environment is the "Centering_AO_system" environment, which is used to center the light of the central star. The second environment is the "Sharpening_AO_system" environment, which is used to sharpen the image. The third environment is the "Sharpening_AO_system_easy" environment, which is a simplified version of the "Sharpening_AO_system" environment. The fourth environment is the "Darkhole_AO_system" environment, which is used to create a dark hole in the image. The "Darkhole_AO_system" environment was not used in our experiments.

4.1.1. CENTERING AO SYSTEM

To control the centering of the light on the central star, the agent has to only control the first two Zernike modes. The action space is therefore 3-dimensional (we also include the 0th-order Zernike mode). Each action is a continuous value between -0.3 and 0.3 in units of radians. The reason for the range of the action space is to avoid divergence of the system. The observation space is the focal image of the telescope. For the resolution we used, the observation space is 48x48 pixels. Higher-resolution images can be used, but this would increase the computational cost of the experiments. The reward function is the negative of the distance between the center of the image and the center of the star. The closer the center of the image is to the center of the star, the higher the reward.

[1] https://stable-baselines.readthedocs.io/en/master/modules/sac.html

[2] https://github.com/ricolandman/gym_ao

[3] https://github.com/johnkou97/gym_ao

4.1.2. Sharpening AO System

The goal of the "Sharpening_AO_system" environment is to learn a control policy that sharpens the image based on the currently obtained image. The observation space is the focal image of the telescope (48x48 pixels) and the reward function is the Strehl ratio of the image. The Strehl ratio is a measure of image sharpness, between 0 and 1. A Strehl ratio of 1 means that the image is perfectly sharp, while a Strehl ratio of 0 means that the image is completely blurred.

We can choose the amplitude of the wavefront RMS error, which is the standard deviation of the wavefront error. The higher the amplitude of the wavefront RMS error, the more distorted the wavefront will be. We can also choose the number of Zernike modes used to correct the wavefront distortion. The higher the number of Zernike modes, the more accurate the correction applied to the wavefront will be. The default amplitude of the wavefront RMS error is 1.7 and the default number of Zernike modes is 20.

4.1.3. Sharpening AO System Easy

This environment is a simplified version of the "Sharpening_AO_system" environment. The observation space and reward function are the same as in the base environment. The key difference is the filtering of the wavefront distortion. In the base environment, the wavefront distortion always includes an infinite number of Zernike modes, as it would be in a real-world scenario. In the easy environment, the wavefront distortion is filtered to include only a finite number of Zernike modes, equal to the number of Zernike modes we use to correct the wavefront. This makes the environment easier to solve, as we can always achieve an optimal value of the Strehl ratio.

The environment has now the unique property that it becomes harder to solve as the number of Zernike modes used to describe the wavefront distortion (and therefore the number of Zernike modes used to correct the wavefront) increases. Because higher-order Zernike modes are less important for the wavefront, this change is more pronounced in the first few Zernike modes and becomes less pronounced as the number of Zernike modes increases. After a certain number of Zernike modes, the "Sharpening_AO_system_easy" environment will essentially become the same as the "Sharpening_AO_system" environment.

4.2. Experimental Setup

The experiments were conducted using the SAC and A2C algorithms (see Section 2.3). The environments were already implemented in the OpenAI Gym framework, however, an additional wrapper was created to allow the environments to be used with the Stable Baselines library. All the different experiments and wrappers can be found in a publicly available repository [4].

We first conducted a set of experiments to test the performance of the SAC and A2C algorithms in the "Centering_AO_system" environment. The goal of the experiments was to test the performance of the algorithms in a simple environment, to ensure that the algorithms were implemented correctly. We varied the size of the buffer used in the SAC algorithm to see how it affected the performance of the algorithm. More specifically, we tested the performance of the SAC algorithm using buffer sizes of 10, 100, and 1000. The number of steps used for training was 100,000. For robustness, we averaged the results over at least 3 runs.

We then worked on the "Sharpening_AO_system_easy" environment. We again used the SAC and A2C agents to train in the environment. We performed a set of experiments using different numbers of Zernike modes to describe and correct the wavefront distortion. We first trained the agents using 2 Zernike modes, then 5, 9, 14, 20, 27 and 35. The number of steps used for training was different for each experiment, as the number of Zernike modes used to describe the wavefront distortion affects the difficulty of the environment. The number of steps used for training was 100,000 for the 2 Zernike modes experiment, 200,000 for the 5 Zernike modes experiment, 200,000 for the 9 Zernike modes experiment, 300,000 for the 14 Zernike modes experiment, 800,000 for the 20 Zernike modes experiment, 2,000,000 for the 27 Zernike modes experiment and 6,000,000 for the 35 Zernike modes experiment. We also varied the buffer size used in the SAC algorithm, increasing it as the number of Zernike modes increased. For the 2 Zernike modes experiment, we used a buffer size of 100, 1000, 10000, and 20000. For 5 Zernike modes, we tested buffer sizes of 1000, 10000, and 20000. We then used buffer sizes of 10000 and 20000 for the 9 Zernike modes experiment, 10000, 20000, and 50000 for the 14 Zernike modes experiment and 10000, 20000, and 50000 for the 20 Zernike modes experiment. Finally, we only used a buffer size of 100000 for the 27 and 35 Zernike modes experiments. Once again, we averaged the results over at least 3 runs for robustness.

Lastly, we experimented with the "Sharpening_AO_system" environment, using the SAC algorithm. We first used the default amplitude of 1.7 for the wavefront RMS error and trained two agents using 14 and 20 Zernike modes. We then lowered the amplitude of the wavefront RMS error to 1.2 and trained a single agent using 20 Zernike modes. For all experiments, we used a buffer size of 100000, which is the maximum buffer size we could use with the computational resources we had. Because of computational constraints, we didn't keep a constant number of steps for training, and in some cases, we only performed a single run.

[4] https://github.com/johnkou97/AdaptiveOptics

In all experiments, we used the default hyperparameters for the SAC and A2C algorithms, as provided by the Stable Baselines library. The only hyperparameter we changed was the buffer size used in the SAC algorithm, as mentioned previously. For each experiment, we tested the performance during training by comparing it to a "no-agent" baseline, which is the performance in the environment without any agent controlling the deformable mirror. This means that the actions at each step are always 0. This was chosen over a random agent as the random agent would give a worse performance because of the divergence of the system.

We kept track of the performance of the agents during training by getting the reward at each step to create a learning curve. We also evaluated the performance of agents after training. For the "Centering_AO_system" environment, we evaluated the performance of the best agent over 1000 episodes, with each episode being 100 steps long. For the "Sharpening_AO_system_easy" environment, we evaluated the performance of each agent over 100 episodes, with each episode being 100 steps long. We then picked the best agent and evaluated its performance over 1000 episodes. In each evaluation, we also evaluated the performance of the "no-agent" baseline. For the "Sharpening_AO_system" environment, we did not evaluate the performance of the agents after training, as the agents' training did not indicate learning beyond the "no-agent" baseline.

5. Results

In this section, we present the results of the experiments that we discussed in the previous section. In Figure 1, we show the learning curves for the "Centering_AO_system" environment. We can see that the SAC algorithm with a buffer size of 1000 was able to learn a good control policy, while the A2C algorithm as well as the SAC algorithm with buffer sizes of 10 and 100 were not able to perform better than the "no-agent" baseline. In Figure 2, we show the evaluation performance of the best agent, which was the SAC algorithm with a buffer size of 1000, compared to the "no-agent" baseline. We can see that the best agent was able to get very close to the optimal value of the reward (which is 0), while the "no-agent" baseline has a much lower performance.

We can then move on to the "Sharpening_AO_system_easy" environment. In Figure 3 we show the learning curves for the different agents trained in the environment using 2 Zernike modes. We can see that all the agents were able to perform better than the "no-agent" baseline, with the SAC performance being better than the A2C performance. The size of the buffer used in the SAC algorithm had a significant effect on the performance of the agent, with the best performance being achieved with a buffer size of 20000. Regarding the evaluation performance on 1000 episodes, we can see in Figure 4 that the best agent, based on the 100-episode evaluation, was able to outperform the "no-agent" baseline in every episode. The performance was also very close to what we would expect in a realistically corrected wavefront. For this case and all other settings of the "Sharpening_AO_system_easy" environment, the evaluation of all agents on 100 episodes can be found in Appendix B.

We then continue with a harder version of the environment, using 5 Zernike modes. In Figure 5 we show the learning curves for the different agents trained in the environment using 5 Zernike modes. We can now see a clear distinction between the performance of the SAC and A2C algorithms, with the SAC algorithm being able to learn a good control policy, while the A2C algorithm was barely able to per-

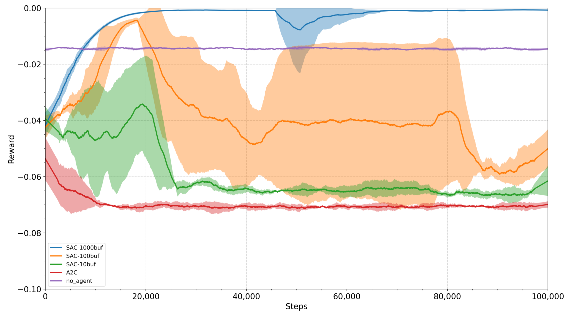

Figure 1. Learning Curves for different agents in the "Centering_AO_system" environment. The x-axis is the number of steps and the y-axis is the reward at each step, which is the negative of the distance between the center of the image and the center of the source. The "no-agent" baseline is the performance of the environment without any agent controlling the deformable mirror and is represented by the purple line.

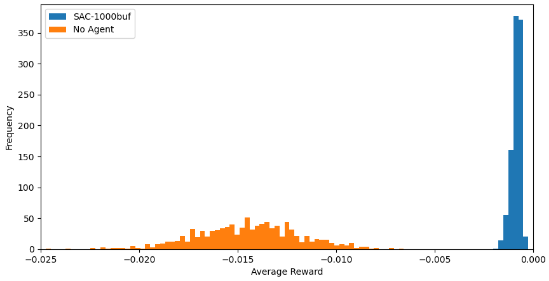

Figure 2. Evaluation Performance of the best agent in the "Centering_AO_system" environment. The x-axis is the average reward over each episode and the y-axis is the frequency of each reward. We compare the performance of the best agent to the "no-agent" baseline.

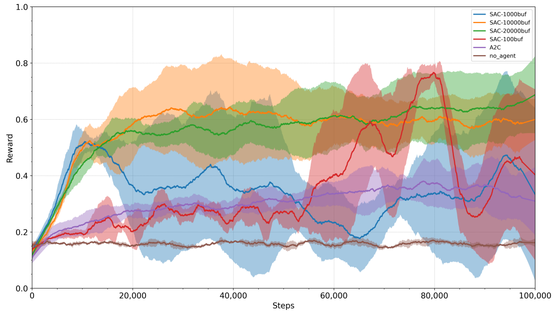

Figure 3. Learning Curves for different agents in the "Sharpening_AO_system_easy" environment using 2 Zernike modes. The x-axis is the number of steps and the y-axis is the reward at each step, which is the Strehl ratio of the image. The "no-agent" baseline is the performance of the environment without any agent controlling the deformable mirror and is represented by the brown line.

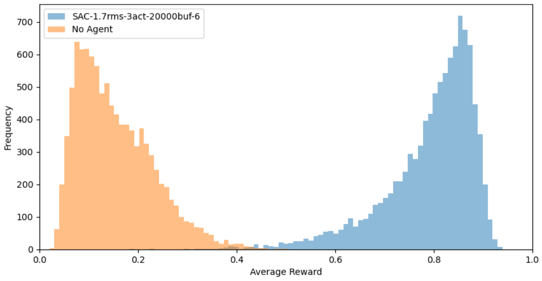

Figure 4. Evaluation Performance of the best agent in the "Sharpening_AO_system_easy" environment using 2 Zernike modes. The x-axis is the average reward over each episode and the y-axis is the frequency of each reward. We compare the performance of the best agent to the "no-agent" baseline.

form better than the "no-agent" baseline in some runs. The size of the buffer used in the SAC algorithm had, again, a significant effect on the performance of the agent. The best performance was achieved with a buffer size of 10000, which had similar performance to the 20000 buffer size, but better stability on different runs. Both of these buffer sizes were able to outperform the 1000 buffer size, which was able to learn, but with a worse performance. On the evaluation performance, we can see in Figure 6 that the 20000 buffer size was able to outperform the "no-agent" baseline, in almost every episode, with a good performance for our standards.

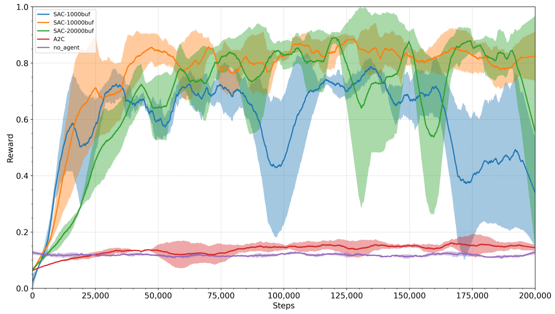

Figure 5. Learning Curves for different agents in the "Sharpening_AO_system_easy" environment using 5 Zernike modes. The x-axis is the number of steps and the y-axis is the reward at each step, which is the Strehl ratio of the image. The "no-agent" baseline is the performance of the environment without any agent controlling the deformable mirror and is represented by the purple line.

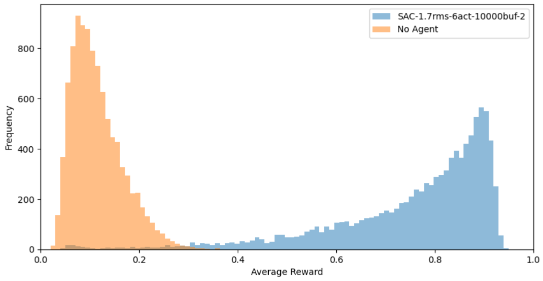

Figure 6. Evaluation Performance of the best agent in the "Sharpening_AO_system_easy" environment using 5 Zernike modes. The x-axis is the average reward over each episode and the y-axis is the frequency of each reward. We compare the performance of the best agent to the "no-agent" baseline.

In Figure 7 we show the learning curves for the different agents trained in the environment using 9 Zernike modes. The A2C algorithm was only able to perform as well as the "no-agent" baseline, while the SAC algorithm was able to learn a good control policy. For the two buffer sizes used in the SAC algorithm, we can see that the 20000 buffer size was able to outperform the 10000 buffer size, with better performance and stability. We also noticed that the 10000 buffer size was able to learn in fewer steps than the 20000 buffer size and reach an acceptable performance, but with worse stability. On the evaluation performance, we can see in Figure 8 that our best agent was able to outperform the "no-agent" baseline, in all but a few episodes, with a good performance for our standards. There is however a big spread in the performance of the agent, and also a spike near the zero reward, which means that in some episodes the agent failed to correct the wavefront.

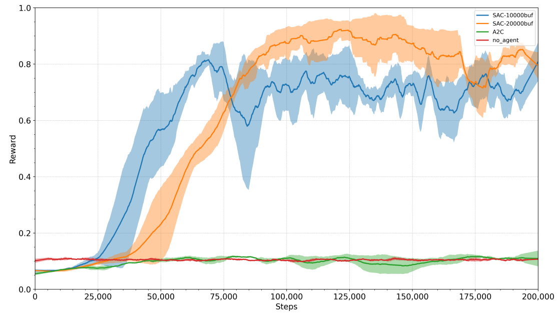

Figure 7. Learning Curves for different agents in the "Sharpening_AO_system_easy" environment using 9 Zernike modes. The x-axis is the number of steps and the y-axis is the reward at each step, which is the Strehl ratio of the image. The "no-agent" baseline is the performance of the environment without any agent controlling the deformable mirror and is represented by the red line.

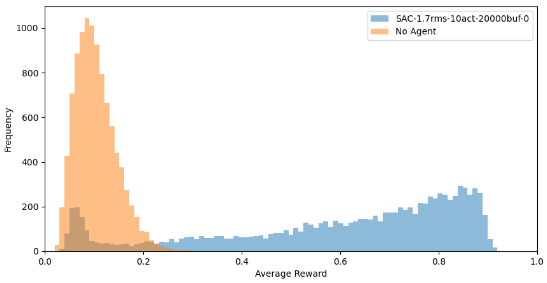

Figure 8. Evaluation Performance of the best agent in the "Sharpening_AO_system_easy" environment using 9 Zernike modes. The x-axis is the average reward over each episode and the y-axis is the frequency of each reward. We compare the performance of the best agent to the "no-agent" baseline.

For the 14 Zernike modes experiment, we show the learning curves for the different agents trained in the environment in Figure 9. The A2C algorithm was barely able to reach the performance of the "no-agent" baseline, while the SAC algorithm was able to learn a good control policy for different buffer sizes. The 10000 buffer size was the worst performing out of the three buffer sizes. The other two buffer sizes had a similar performance, with both showing some instability in the later stages of training. Based on the evaluation performance of all the different agents with the baseline in Figure 21, we have chosen the 20000 buffer size as the best agent. In Figure 10 we show the evaluation performance of this agent, which was able to outperform the "no-agent" baseline, with a very similar shape as in Figure 8 for the 9 Zernike modes experiment.

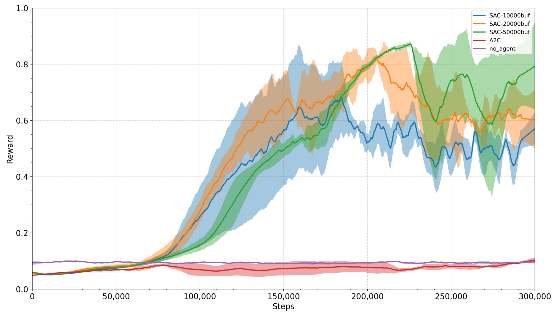

Figure 9. Learning Curves for different agents in the "Sharpening_AO_system_easy" environment using 14 Zernike modes. The x-axis is the number of steps and the y-axis is the reward at each step, which is the Strehl ratio of the image. The "no-agent" baseline is the performance of the environment without any agent controlling the deformable mirror and is represented by the purple line.

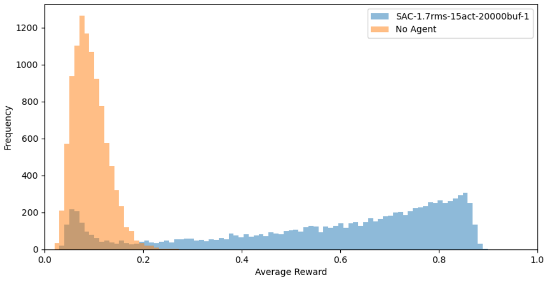

Figure 10. Evaluation Performance of the best agent in the "Sharpening_AO_system_easy" environment using 14 Zernike modes. The x-axis is the average reward over each episode and the y-axis is the frequency of each reward. We compare the performance of the best agent to the "no-agent" baseline.

Making the environment even harder, we show the learning curves for the different agents trained in the environment using 20 Zernike modes in Figure 11. In this case, the A2C algorithm was not able to perform better than the "no-agent" baseline, while the SAC performed well, especially with higher buffer sizes. We, however, notice the instability of the 100000 buffer size, which in one run completely "forgot" the learned policy at the later stages of training. Despite that, it also provided the best performance, and we have chosen it as the best agent. In Figure 12 we show the evaluation performance of this agent, with clearly better performance than the "no-agent" baseline. It is clear, though, that in many episodes the agent failed to correct the wavefront, with a reward of close to zero.

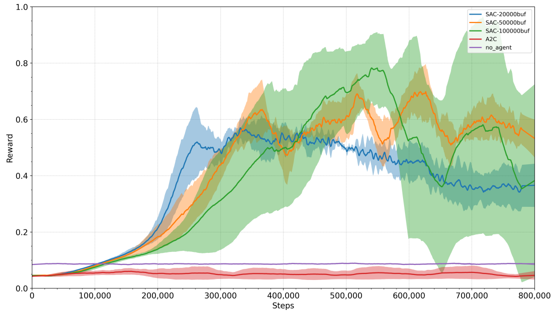

Figure 11. Learning Curves for different agents in the "Sharpening_AO_system_easy" environment using 20 Zernike modes. The x-axis is the number of steps and the y-axis is the reward at each step, which is the Strehl ratio of the image. The "no-agent" baseline is the performance of the environment without any agent controlling the deformable mirror and is represented by the purple line.

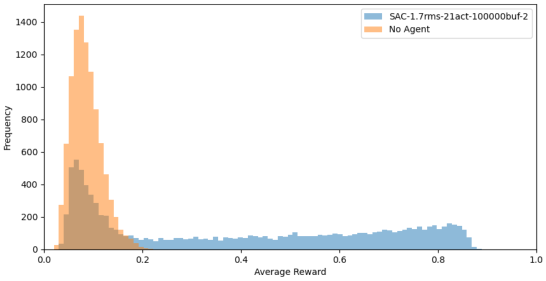

Figure 12. Evaluation Performance of the best agent in the "Sharpening_AO_system_easy" environment using 20 Zernike modes. The x-axis is the average reward over each episode and the y-axis is the frequency of each reward. We compare the performance of the best agent to the "no-agent" baseline.

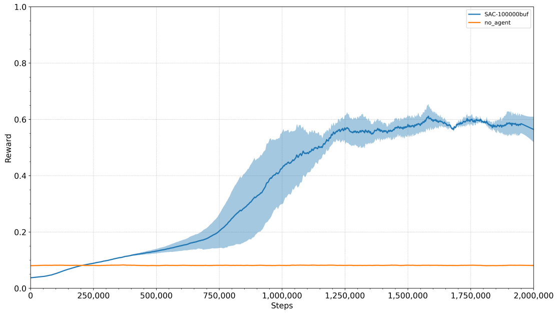

Figure 13. Learning Curve for the agent in the "Sharpening_AO_system_easy" environment using 27 Zernike modes. The x-axis is the number of steps and the y-axis is the reward at each step, which is the Strehl ratio of the image. The "no-agent" baseline is the performance of the environment without any agent controlling the deformable mirror and is represented by the orange line.

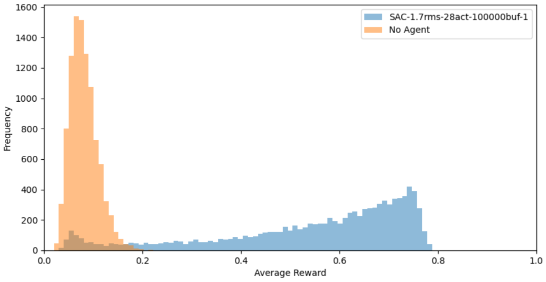

Figure 14. Evaluation Performance of the agent in the "Sharpening_AO_system_easy" environment using 27 Zernike modes. The x-axis is the average reward over each episode and the y-axis is the frequency of each reward. We compare the performance of the agent to the "no-agent" baseline.

In the next step, we increased the number of Zernike modes to 27. The learning curves are shown in Figure 13. We now have only one agent, which is the SAC algorithm with a buffer size of 100000. The agent was able to learn a good control policy, with a better performance than the "no-agent" baseline. The evaluation performance of this agent is shown in Figure 14, with a good performance, but a lower ceiling than in the previous experiments. The spread, however, is smaller than in the previous experiment, and the same is true for the spike near the zero reward.

The final experiment in the "Sharpening_AO_system_easy" was done with 35 Zernike modes. In this version, we had a clear distinction from all the previous experiments as we were no longer able to see any learning with the agent we used, which was, once again, the SAC algorithm with a 100000 buffer size. The learning curve didn't even reach the "no_agent" baseline and we therefore didn't evaluate it, as it was clearly below optimal. We show the learning curve in Figure 15.

Moving from the "Sharpening_AO_system_easy" to the "Sharpening_AO_system" environment, we had an idea of what to expect, since this is a harder environment to train on even when compared to using 35 Zernike modes on the easy version. We made two attempts with different RMS errors for the wavefront, and in both cases, we were not able to achieve any learning. We show the learning curves for the 1.7 RMS error in Figure 16 and for the 1.2 RMS error in Figure 17. We performed only a single run for each agent

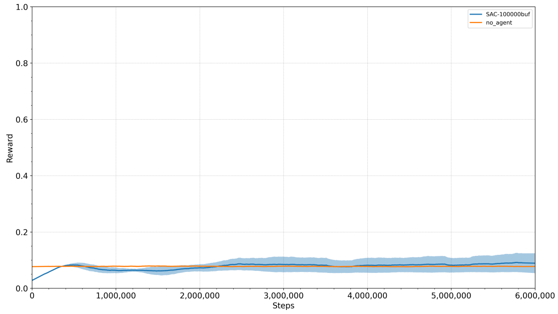

Figure 15. Learning Curve for the agent in the "Sharpening_AO_system_easy" environment using 35 Zernike modes. The x-axis is the number of steps and the y-axis is the reward at each step, which is the Strehl ratio of the image. The "no-agent" baseline is the performance of the environment without any agent controlling the deformable mirror and is represented by the orange line.

in the 1.7 RMS error experiment and two runs for the 1.2 RMS error experiment. We didn't evaluate the performance of the agents at the end of training, as the agents were not able to learn a good control policy.

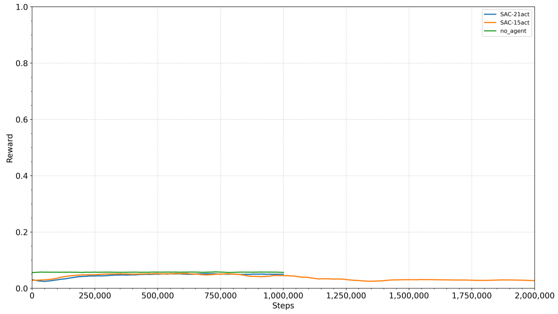

Figure 16. Learning Curves for the agent in the "Sharpening_AO_system" environment using a 1.7 RMS error. One agent was trained using 14 Zernike modes and another agent was trained using 20 Zernike modes to control the AO system. The x-axis is the number of steps and the y-axis is the reward at each step, which is the Strehl ratio of the image. The "no-agent" baseline is the performance of the environment without any agent controlling the deformable mirror and is represented by the green line.

From all the different experiments we run it becomes clear that the SAC agent is capable of handling an AO control system without the need for a wavefront sensor. We were able to successfully train the SAC agent on the "Centering_AO_system" as well as different versions of the "Sharpening_AO_system_easy". The results were suboptimal in the sense that we would expect them to be more stable in the evaluation phase, which could be attributed to the lack of hyperparameter tuning.

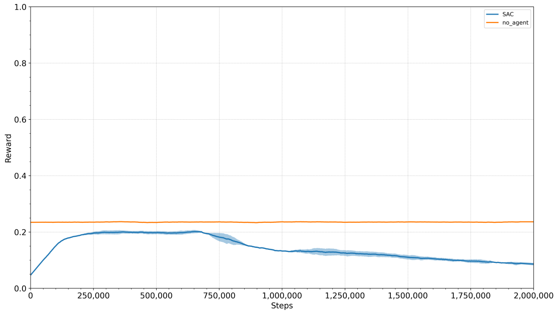

Figure 17. Learning Curve for the agent in the "Sharpening_AO_system" environment using 20 Zernike modes and a 1.2 RMS error. The x-axis is the number of steps and the y-axis is the reward at each step, which is the Strehl ratio of the image. The "no-agent" baseline is the performance of the environment without any agent controlling the deformable mirror and is represented by the orange line.

The main issue was the lack of any learning when we switched to the 35 Zernike modes version of the "Sharpening_AO_system_easy", which of course continued when we experimented with the "Sharpening_AO_system" environment. Such sophisticated environments require some additional tricks or a more careful consideration of the Neural Networks being used and the hyperparameters that control the learning process. We discuss those aspects more in-depth in the next section.

6. Discussion

In this section, we critically reflect on our work and discuss its shortcomings as well as implications for future work in the field.

While the results presented in Section 5 show that our experiments were successful to a certain degree, we see two main shortcomings of our work. First, we applied our chosen RL agents almost completely out of the box. We did not perform systematic hyperparameter tuning, nor model parameter tuning except for trying different buffer sizes. While the results do show successful learning, the lack of parameter tuning prohibits us from understanding the full potential these RL agents could unfold with their (close to) optimal parameter configuration. This is due to the extreme effect even a single parameter can have on an agent's performance, as we demonstrated for the buffer size. Second, our results lack a competitive baseline based on previous work.

Instead, we only compare our results to an inactive agent that never performs any actions. Comparison to previously proposed RL-based AO controllers would allow for a better understanding of how our approach integrates into the existing work. While problematic for our work, we also see this as a challenge for the whole AO-control research field as there is no coherent evaluation, and proposed methods are not compared to competitive methods. Instead, the most commonly used baseline is an integrator which is a simple control algorithm that based control on accumulations of wavefront sensor measurements over time. Since we perform sensorless control, an integrator was not a suitable baseline for us.

The lack of a universal evaluation framework and applicable baselines led to us comparing our results to that of an inactive agent. However, the most limiting aspect of our experiments was the computational cost together with the project timeframe. It is due to these reasons that we decided to not perform in-depth parameter tuning. We tried to reduce the computational load by applying the SR-SAC agent which was proposed as a more sample-efficient agent (D'Oro et al., 2023). We hoped this sample efficiency would reduce training time and, therefore, allow running more and longer experiments. However, we could not achieve any learning using the SR-SAC implementation provided by its original authors [5]. A more detailed summary of our work using SR-SAC can be found in Appendix A.

Given our promising results and the discussed shortcomings of our work, we see a few avenues for future work building up on and going beyond our work. Regarding the discussed limitations of our work's shortcomings, we see value in future work overcoming them. For the parameter tuning, this could simply be achieved by investing more computational resources allowing for a systematic and extensive parameter tuning procedure. In addition to tuning hyper and model parameters, we also think an extensive comparison of available pre-implemented RL agents could already provide valuable insights into the suitability of different agents for the given task without requiring too much implementation effort. The fact that, despite only considering two agents and performing almost no parameter tuning, our results show successful learning for up to 27 Zernike modes causes us to be optimistic that parameter tuning and comparison of more agents would allow for significant performance improvements and enable learning even for more than 27 Zernike modes. Tackling the issue of comparability between our method and previously proposed ones is less straightforward. It requires either identifying an applicable previously proposed method whose implementation is available and which can serve as a competitive baseline for our environments, or the development of a general benchmark for evaluating sensorless AO control for astronomical imaging. While the former would be the simpler way to improve our work, we see the latter as much more impactful for tackling the issue of comparability beyond our work and for the whole research field.

7. Conclusion

In this work, we have demonstrated that a model-free SAC agent can successfully learn AO control for the centering or sharpening of astronomical images. Contrary to prior work on AO control, our agent learns directly from the resulting images without requiring measurements from a wavefront sensor. Due to this sensorless approach, we can skip the step of predicting actions' effects on sensor measurements and can train directly for their effect on the acquired image. Our experiments confirmed that the large action space caused by a large number of individual mirrors within a single telescope poses a significant challenge for applying RL to AO control. Nonetheless, our agent was able to show successful learning for up to 27 Zernike modes. By overcoming the discussed shortcomings of our work, we see large potential in approaching similar performance also in more realistic settings with much larger action spaces. For now, we see our work as a promising first step to achieving RL-based sensorless AO control for astronomical imaging.

References

Beckers, J. M. Adaptive Optics for Astronomy: Principles, Performance, and Applications. ARA&A, 31:13–62, January 1993. doi: 10.1146/annurev.aa.31.090193.000305.

Davies, R. and Kasper, M. Adaptive Optics for Astronomy. ARA&A, 50:305–351, September 2012. doi: 10.1146/annurev-astro-081811-125447.

D'Oro, P., Schwarzer, M., Nikishin, E., Bacon, P., Bellemare, M. G., and Courville, A. C. Sample-efficient reinforcement learning by breaking the replay ratio barrier. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=OpC-9aBBVJe.

Durech, E., Newberry, W., Franke, J., and Sarunic, M. V. Wavefront sensor-less adaptive optics using deep reinforcement learning. *Biomed. Opt. Express*, 12(9):5423–5438, Sep 2021. doi: 10.1364/BOE.427970. URL https://opg.optica.org/boe/abstract.cfm?URI=boe-12-9-5423.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Dy, J. G. and Krause, A. (eds.), *Proceedings of*

[5] https://github.com/proceduralia/high_replay_ratio_continuous_control

the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, volume 80 of *Proceedings of Machine Learning Research*, pp. 1856–1865. PMLR, 2018. URL http://proceedings.mlr.press/v80/haarnoja18b.html.

Ke, H., Xu, B., Xu, Z., Wen, L., Yang, P., Wang, S., and Dong, L. Self-learning control for wavefront sensorless adaptive optics system through deep reinforcement learning. *Optik*, 178:785–793, 2019. ISSN 0030-4026. doi: https://doi.org/10.1016/j.ijleo.2018.09.160. URL https://www.sciencedirect.com/science/article/pii/S0030402618314785.

Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Sallab, A. A. A., Yogamani, S., and Pérez, P. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23 (6):4909–4926, 2022. doi: 10.1109/TITS.2021.3054625.

Landman, R., Haffert, S. Y., Radhakrishnan, V. M., and Keller, C. U. Self-optimizing adaptive optics control with reinforcement learning for high-contrast imaging. *Journal of Astronomical Telescopes, Instruments, and Systems*, 7(3):039002–039002, 2021.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In Balcan, M. and Weinberger, K. Q. (eds.), *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pp. 1928–1937. JMLR.org, 2016. URL http://proceedings.mlr.press/v48/mniha16.html.

Nousiainen, J., Rajani, C., Kasper, M., and Helin, T. Adaptive optics control using model-based reinforcement learning. *Optics Express*, 29(10):15327–15344, 2021.

Parvizi, P., Zou, R., Bellinger, C., Cheriton, R., and Spinello, D. Reinforcement learning-based wavefront sensorless adaptive optics approaches for satellite-to-ground laser communication, 2023a.

Parvizi, P., Zou, R., Bellinger, C., Cheriton, R., and Spinello, D. Reinforcement learning environment for wavefront sensorless adaptive optics in single-mode fiber coupled optical satellite communications downlinks. *Photonics*, 10(12), 2023b. ISSN 2304-6732. doi: 10.3390/photonics10121371. URL https://www.mdpi.com/2304-6732/10/12/1371.

Pou, B., Ferreira, F., Quinones, E., Gratadour, D., and Martin, M. Adaptive optics control with multi-agent model-free reinforcement learning. *Opt. Express*, 30(2):2991–3015, Jan 2022. doi: 10.1364/OE.444099. URL https://opg.optica.org/oe/abstract.cfm?URI=oe-30-2-2991.

Roddier, F. *Adaptive Optics in Astronomy*. Cambridge University Press, 2004.

A. Experiments with SR-SAC

A large part of our work focused on applying a Scaled-by-Resetting SAC (SR-SAC) agent, a variant of the SAC agent that would allow for training with a higher sample efficiency (D’Oro et al., 2023). The idea behind SR-SAC is to perform periodic resets of the agent’s critic and actor parameters. These resets are introduced to reduce the capacity loss which refers to artificial neural networks losing their ability to generalize to new unseen data when being trained for too long. This is especially problematic for RL as the data, i.e. collected experiences, changes based on the state of training, i.e. how good the current policy is. Overcoming this issue of capacity loss then allowed for the use of higher replay ratios, i.e. more training can be done using experiences from the replay buffer without environment interaction. It is this increase in replay ratio that allows SR-SAC to be more sample efficient.

However, we were not able to successfully train the SR-SAC agent on any of our used environments. In fact, even when turning off the resets, making it equivalent to the normal SAC agent, and using the same parameter as for our SAC agent, we were not able to observe any learning. This is despite us being able to successfully replicate results on the DeepMind Control Suite benchmark reported by D’Oro et al. (2023).

As a consequence, we discarded the SR-SAC agent and focused on evaluating the SAC and A2C agents available through stable baselines. Nonetheless, we still believe that computationally efficient variants of SAC or other agents could be a valuable option for future research as we have seen that increasing action dimensions causes significantly longer training times. More efficient agents could reduce the overall time needed for training and, thus, free resources for parameter tuning.

B. Extra Evaluation Figures

We provide additional evaluation figures for the experiments conducted in the “Sharpening_AO_system_easy” environment. The figures show the evaluation performance of all the different agents at the end of training, as well as the performance of the “no-agent” baseline. For this evaluation, we used 100 episodes, with each episode being 100 steps long. The reward for each episode is the average reward over the 100 steps.

We show the evaluation performance of the agents trained using 2 Zernike modes in Figure 18, 5 Zernike modes in Figure 19, 9 Zernike modes in Figure 20, 14 Zernike modes in Figure 21, and 20 Zernike modes in Figure 22.

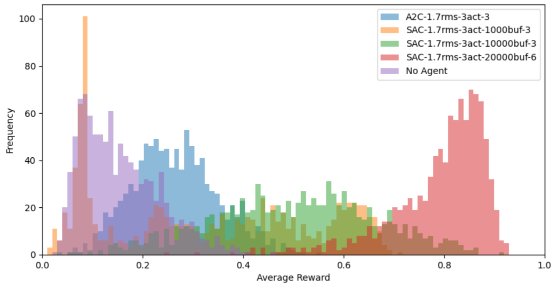

Figure 18. Evaluation Performance of all the different agents in the “Sharpening_AO_system_easy” environment using 2 Zernike modes. The x-axis is the average reward over each episode and the y-axis is the frequency of each reward. We compare the performance of the different agents to the “no-agent” baseline.

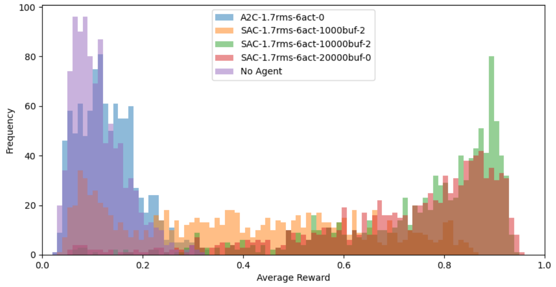

Figure 19. Evaluation Performance of all the different agents in the “Sharpening_AO_system_easy” environment using 5 Zernike modes. The x-axis is the average reward over each episode and the y-axis is the frequency of each reward. We compare the performance of the different agents to the “no-agent” baseline.

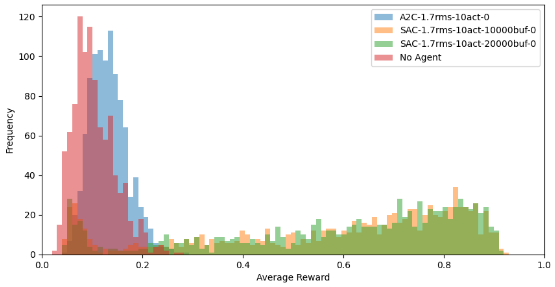

Figure 20. Evaluation Performance of all the different agents in the “Sharpening_AO_system_easy” environment using 9 Zernike modes. The x-axis is the average reward over each episode and the y-axis is the frequency of each reward. We compare the performance of the different agents to the “no-agent” baseline.

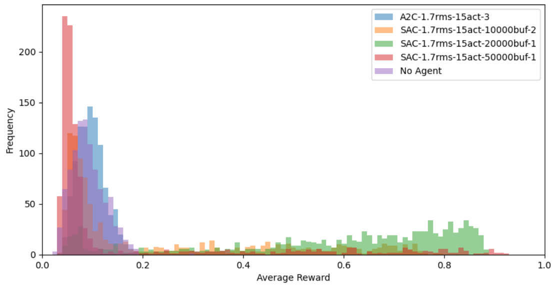

Figure 21. Evaluation Performance of all the different agents in the “Sharpening_AO_system_easy” environment using 14 Zernike modes. The x-axis is the average reward over each episode and the y-axis is the frequency of each reward. We compare the performance of the different agents to the “no-agent” baseline.

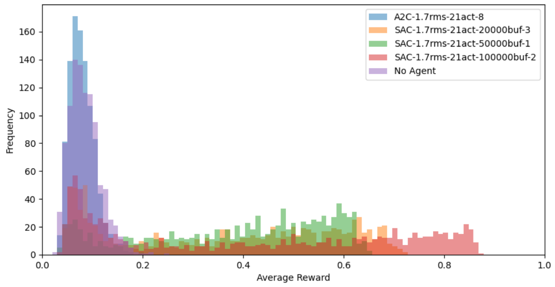

Figure 22. Evaluation Performance of all the different agents in the “Sharpening_AO_system_easy” environment using 20 Zernike modes. The x-axis is the average reward over each episode and the y-axis is the frequency of each reward. We compare the performance of the different agents to the “no-agent” baseline.