Self-Driving Telescopes: Autonomous Scheduling of Astronomical Observation Campaigns with Offline Reinforcement Learning

Franco Terranova
Department of Information Engineering
University of Pisa
Pisa, IT 56126
`terranovafr@icloud.com`

M. Voetberg
Computational Science and AI Directorate
Fermi National Accelerator Laboratory
Batavia, IL 60510
`maggiev@fnal.gov`

Brian Nord
Fermi National Accelerator Laboratory;
Kavli Institute for Cosmological Physics
Department of Astronomy and Astrophysics,
University of Chicago;
`nord@fnal.gov`

Amanda Pagul
Space Telescope Science Institute
Baltimore, MD 21218
`apagul@stsci.edu`

Abstract

Modern astronomical experiments are designed to achieve multiple scientific goals – from studies of galaxy evolution to cosmic acceleration. These goals require data of many different classes of night-sky objects, each of which has a particular set of observational needs. These observational needs are typically in strong competition with one another. This poses a challenging multi-objective optimization problem that remains unsolved. The effectiveness of Reinforcement Learning (RL) as a valuable paradigm for training autonomous systems has been well-demonstrated, and it may provide the basis for self-driving telescopes capable of optimizing the scheduling for astronomy campaigns. Simulated datasets containing examples of interactions between a telescope and a discrete set of sky locations on the celestial sphere can be used to train an RL model to sequentially gather data from these several locations to maximize a cumulative reward as a measure of the quality of the data gathered. We use simulated data to test and compare multiple implementations of a Deep Q-Network (DQN) for the task of optimizing the schedule of observations from the Stone Edge Observatory (SEO). We combine multiple improvements on the DQN and adjustments to the dataset, showing that DQNs can achieve an average reward of 87%±6% of the maximum achievable reward in each state on the test set. This is the first comparison of offline RL algorithms for a particular astronomical challenge and the first open-source framework for performing such a comparison and assessment task.

1 Introduction

Scheduling observations for telescopes is a challenging task, known to be an NP-hard (Non-deterministic polynomial-time hard) problem. This level of complexity has also been highlighted in other studies (1). Optimizing telescope scheduling can significantly enhance the efficiency of the planning process and the effectiveness of surveys, which may be otherwise sub-optimal in the case of manual optimization due to the complexity of the problem. We have developed and implemented a

framework based on the PTAN (PyTorch AgentNet) library (2) of RL tools and utilities to compare various RL algorithms to achieve this goal. This framework can be used on any offline dataset – i.e., data typically generated by an expert, a simulation, or a historical record of observations. The offline dataset must contain observation variables, actions taken in response to those observations, and examples of the resulting data quality after these actions. The dataset is scanned and used by our framework to achieve an optimized observing schedule. This study focuses on the results obtained from an offline simulation dataset based at the Stone Edge Observatory (SEO) [1]. When dealing with an offline dataset, which has a limited amount of data, value-based methods tend to perform better than other methods since they are sample-efficient and well-suited for off-policy learning. (3) This consideration has been confirmed with our specific dataset, on which they overcome other classes of methods, and this is the reason Deep Q-Networks (DQNs) (4) are the main targets of this work.
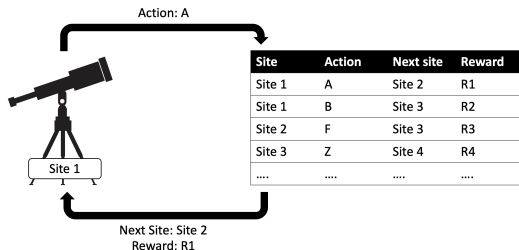

Figure 1: The RL paradigm applied to our task considers the telescope agent interacting with the environment represented by an offline dataset. The RL agent gathers the state, and executes an action, and the environment will respond with a reward and a new state.

1.1 Limitations

The analysis performed in this study can be biased by the characteristics of the specific dataset used in terms of generalization since it represents a specific simulation survey of a specific observatory. However, the framework we developed for this study can be used to compare multiple RL algorithms on any offline simulation dataset and can be used in the future to study the capacity for generalization of algorithms regardless of the offline dataset used. Our simulated dataset contains only positions of sky objects, but not factors such as weather or climate conditions. Furthermore, when addressing multi-objective problems, optimizing a single data quality metric, as will be performed in this study, may fail to fully represent the problem, even if it considers a general metric for data quality. A more sophisticated reward function should be needed to achieve a specific objective.

1.2 Related work

In the field of self-driving telescopes and schedule optimization, most of the research has been primarily approached with heuristic methods, as demonstrated in prior work (5) (6). In the study by Alba et al. (7), it was observed that tabular approaches had their limitations in this context. On the other hand, Jia et al. (8) primarily focused on the challenges and solutions related to distributed telescope arrays. More recently, automated approaches have been tested for large cosmic surveys: e.g., RL algorithms for the Rubin Observatory's Legacy Survey of Space and Time (LSST) (e.g., 9) and graph neural networks for spectroscopic surveys (e.g., 10). All of these classes of techniques require an environment, in the real world or otherwise, to be trained.

2 Methodology

We describe the methodology we built to address the application of value-based RL techniques on the benchmark SEO offline dataset, which contains a set of discrete locations in the sky. Consequently, it requires a discrete action space. Datasets, checkpoints, and the code used are all available in Zenodo (11). For each algorithm and experiment, we make available the tensorboard logs and

---

[1]SEO website: `https://sites.google.com/a/starsatyerkes.net/yerkesprojects/tools/telescope-upgrades/stone-edge-observatory?pli=1`

hyper-parameters. Hyper-parameters related to the problem and algorithms were selected based on discussion with domain experts and after manual comparison.

2.1 Offline dataset

This analysis uses an offline dataset that contains examples of interactions between the SEO telescope and the sky, which are generated through a simulation tool. The offline dataset consists of 309,599 observations, representing a collection of state-action-reward-next state samples. The data representing the observation space (i.e. state space) consists of several telemetry and positioning factors related to the telescope, the sun, and the moon. The action space consists of Right Ascension and Declination coordinates. In particular, the discrete set of unique actions available in the dataset corresponds to the sets of locations available for optimization. The effective exposure time ($t_{effective}$)(12) is the reward variable for the RL agent, and it represents the depth (i.e., quality) of the images collected (Equation 1). This variable indicates the level of brightness required for detecting the faintest point source at a given reference signal-to-noise ratio.

$$T_{eff} \equiv \eta^2 \left(\frac{0.9''}{FWHM} \right) \left(\frac{b_{dark}}{b} \right) * \tau_{exposure} \qquad (1)$$

The equation considers the key factors affecting exposure quality, including seeing (*FWHM*, blurring caused by atmospheric and instrumental factors), atmospheric transmission (η, the fraction of light reaching the instrument), and the sky brightness (*b*), including a component related to low background light levels (b_{dark}). The component $\tau_{exposure}$ is the exposure time of a given observation. The selection of this reward model aims to drive the telescope to prioritize the optimization of the gathered data's quality.

2.2 Action selection

The dataset only contains rewards for a subset of all unique actions in different states. Therefore, the agent might suggest telescope coordinates for which the reward is not available in the dataset in a certain state. Instead of dedicating an output neuron of the neural network for each unique action, which may be chosen in some states in which it will not be available, the dataset's right ascension and declination ranges have been evenly divided into 19 bins, and an output neuron is dedicated to each pair of bins. The training buffer hence selects the action (sky location) closest available – by angular distance – to the chosen action. This number of bins has been properly computed based on the dataset, in order to ensure that there's at least one output neuron mapped to each action, even in the worst-case scenario where all actions are available. This prevents the need to modify the network structure if the dataset expands.

2.3 Evaluation metric: the average effectiveness distribution

During testing, the reward distribution for each state is normalized to a range of [0, 1] to enable fair comparisons: different states have rewards with different scales. We refer to the average reward in an episode as "average effectiveness," and we calculated an average effectiveness distribution by conducting numerous episodes in both the full environment (1000 episodes) and the test set (500 episodes) to compare agents, ensuring result stability for agent comparisons. This distribution is generated using the combination of checkpoints that yielded the best average reward on the training and validation sets.

2.4 DQN Network Architecture

The number of neurons in the first layer of the DQN reflects the number of variables in the observation space (26 neurons). The architecture has three hidden layers (with 64, 128, and 256 neurons, respectively). The LeakyReLU activation function has been used with the amount of leak α set to 0.01. The number of bins was calculated as described in Section 2.2, resulting in 19 bins for the two action variables, hence a 361-dimensional (19x19) output space representing all possible pairs of bins.

3 Results

3.1 Hyper-parameters and normalization

The neural network was trained for 300,000 iterations on three parallel training environments. The selections of hyper-parameters were the result of manual optimization and domain knowledge. The replay buffer's size was set to 5,000 transitions, with a target network's synchronization period of 2,000 iterations. For the exploration strategy, the Basic DQN includes an ϵ-greedy strategy with a linear decay in 100,000 iterations from 1 to 0.05. The holdout method is used with test and validation ratios set to 20%. The combination of z-score normalization for the observation space variables and the intra-state reward distribution pushed the average effectiveness distribution mean value from 36%±24% to a value of 77%±12% on the test set for Basic DQN, demonstrating to be the most effective strategy to improve performances. Several hyper-parameters have also been varied, proving their impact on the performance of the network. In particular, the discounting factor (γ), set at 0.8, the learning rate at 0.001, and a batch size of 128 emerged as the most influential hyperparameters. It is crucial to highlight that the performance of a baseline random agent, which selects a random site within each state, yields an average effectiveness distribution with a mean value of 39%±12%.

3.2 Improvements in literature

Further advanced modifications in the literature, including Dueling DQN (13), N-steps Bellman unrolling (14), and noisy layers exploration strategy (15), have been compared to highlight the improvements in the performance of the DQN model (Figure 2). These enhancements have been combined, analogously to the Rainbow DQN approach (16). Figure 3 summarizes the comparison between the combination of the DQN improvements, or "Rainbow" DQN, and the original or "Basic" DQN, showing an improvement in general performance, convergence, and generalization capabilities during training.
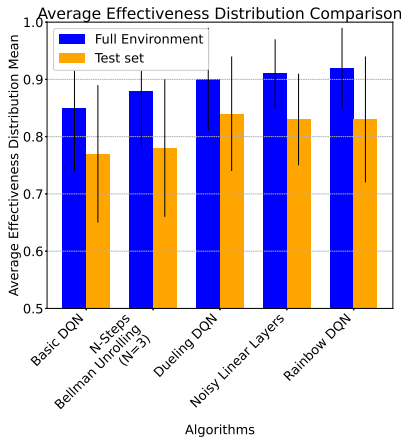

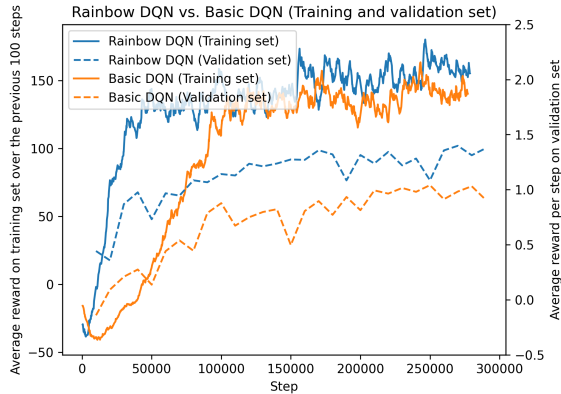Figure 2: Comparison among improvements to the DQN method in terms of average effectiveness distribution.

Figure 3: Average reward trend on the training set during the last 100 iterations and validation average reward per step during training: comparison of the Rainbow DQN and the Basic DQN.

3.3 Evaluation

The Rainbow DQN has been assessed with a K-Fold Cross Validation (K-Fold CV) using K = 5, highlighting how the performances are pretty similar regardless of the portion of the dataset used for training, validating, or testing (Figure 4). Moreover, the robustness of the model has been addressed by decreasing the quantity of samples in the training set. The resulting models trained with a reduced sample size have been compared on the same test set (Figure 5), showcasing the model's capacity to learn effectively even with a lower number of samples, important for the particular context of learning from an offline dataset.
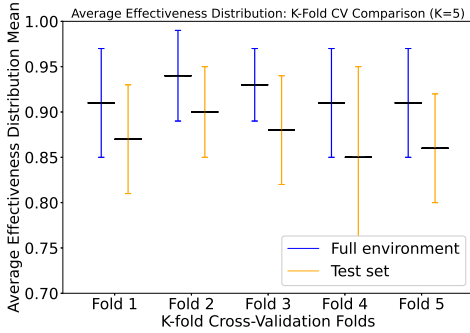
Figure 4: Rainbow DQN K-Fold CV average effectiveness distribution: comparison among folds results.
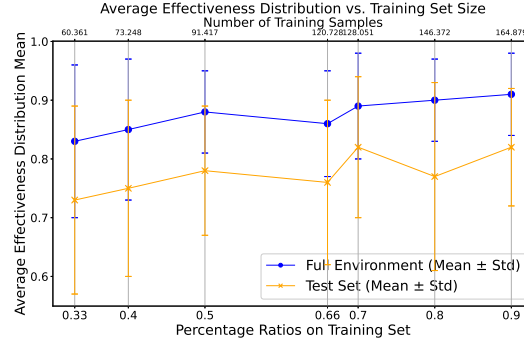

Figure 5: Average effectiveness distribution comparison: reduction of the percentage ratios of the training set.

4   Conclusions and Impact Statement

Our results suggest that RL algorithms can be used to optimize the sequential schedule of a telescope survey. A proper framework has been developed for the purpose of wrapping offline datasets with the goal of using them as environments. Value-based methods, and in particular DQNs, have shown remarkable success in tackling schedule optimization based on an offline simulation dataset located at the SEO. The combination of normalization techniques, hyperparameters exploration, and extensions to the DQN method have pushed the performances of the agent way further. This research is expected to yield substantial advancements in the realm of enhancing telescope observations using RL. Even though this work primarily focuses on the domain of astronomical observations, the methodology discussed could be adapted to any problem related to schedule optimization.

References

[1] A. Hinze, F. Hauke, and T. Schildknecht, "Optimal scheduling for geosynchronous space object follow-up observations using a genetic algorithm." Presented at the AMOS (Advanced Maui Optical and Space Surveillance Technologies Conference) 2016 Conference, 2016.

[2] Shmuma and contributors, "Ptan (python training for agent-based networks)." `https://github.com/Shmuma/ptan`. Version 0.6, MIT License.

[3] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. The MIT Press, 2018.

[4] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *NIPS Deep Learning Workshop 2013, arXiv preprint arXiv:1312.5602*, 2013.

[5] S. Minton, M. D. Johnston, A. B. Philips, and P. Laird, "Solving large-scale constraint satisfaction and scheduling problems using a heuristic repair method," in *AAAI'90: Proceedings of the eighth National conference on Artificial intelligence - Volume 1*, pp. 17–24, July 1990.

[6] E. Naghib, R. J. Vanderbei, and C. Stubbs, "Feature-based telescope scheduler," *PROCEEDINGS OF SPIE*, 2016. SPIEDigitalLibrary.org/conference-proceedings-of-spie.

[7] A. F. A. Hernández, "Sky surveys scheduling using reinforcement learning," 2019. Graduate Research Theses Dissertations. 6789.

[8] P. Jia, Q. Jia, T. Jiang, and J. Liu, "Observation strategy optimization for distributed telescope arrays with deep reinforcement learning," May 2023. DOI: 10.13140/RG.2.2.17173.60641.

[9] E. Naghib, P. Yoachim, R. J. Vanderbei, A. J. Connolly, and R. L. Jones, "A Framework for Telescope Schedulers: With Applications to the Large Synoptic Survey Telescope," *AJ*, vol. 157, p. 151, Mar. 2019. arXiv:1810.04815 [astro-ph].

[10] M. Cranmer, P. Melchior, and B. Nord, "Unsupervised Resource Allocation with Graph Neural Networks," June 2021. arXiv:2106.09761 [astro-ph].

[11] "Datasets, Checkpoints, and Code for "Self-Driving Telescopes: Autonomous Scheduling of Astronomical Observation Campaigns with Offline Reinforcement Learning," Sept. 2023. DOI: 10.5281/zenodo.8388483.

[12] E. J. Neilsen, J. T. Annis, H. T. Diehl, M. E. Swanson, C. D'Andrea, S. Kent, and A. Drlica-Wagner, "Dark energy survey's observation strategy, tactics, and exposure scheduler," 11 2019.

[13] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *Proceedings of The 33rd International Conference on Machine Learning* (M. F. Balcan and K. Q. Weinberger, eds.), vol. 48 of *Proceedings of Machine Learning Research*, (New York, New York, USA), pp. 1995–2003, PMLR, 20–22 Jun 2016.

[14] R. Sutton, "Learning to predict by the methods of temporal differences," *Machine learning*, vol. 3, no. 1, pp. 9–44, 1988.

[15] M. Fortunato, M. G. Azar, B. Piot, J. Menick, I. Osband, A. Graves, V. Mnih, R. Munos, D. Hassabis, O. Pietquin, C. Blundell, and S. Legg, "Noisy networks for exploration," 2019.

[16] M. Hessel, J. Modayil, H. van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver, "Rainbow: Combining improvements in deep reinforcement learning," 2017.

[17] "Stone edge observatory website." https://sites.google.com/a/starsatyerkes.net/yerkesprojects/tools/telescope-upgrades/stone-edge-observatory. Accessed: July 5, 2023.

[18] M. Lapan, *Deep Reinforcement Learning Hands-On*. Packt Publishing, 2018.

[19] S. Ravichandiran, *Deep Reinforcement Learning with Python*. Packt Publishing, 2020.

[20] E. Neilsen and contributors, "Skybright." https://github.com/ehneilsen/skybright. MIT License.

[21] A. et al., "The Astropy Project: Sustaining and Growing a Community-oriented Open-source Project and the Latest Major Release (v5.0) of the Core Package," , vol. 935, p. 167, Aug. 2022. 3-clause BSD style license.

[22] C. J. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, pp. 279–292, 1992.