An exploration into the effects of discount factor and entropy coefficient on the stability of PPO and A2C

Revan Murton
Falmouth University
https://github.falmouth.ac.uk/RM305181/COMP213-305181

***Abstract*-This paper investigates the stability of Proximal Policy Optimization and Advantage Actor-Critic algorithms when either entropy coefficient and discount factor is varied. Utilizing data derived from the stable-baselines3 framework and the Gymnasium environment "Lunar Lander," the analysis reveals that A2C demonstrates significantly higher stability under either of these conditions, challenging the initial hypothesis. This observation suggests the presence of a stabilizing mechanism within A2C that mitigates the loss of stability typically expected when these hyperparameters are modified. Conversely, it raises questions about the efficacy of PPO's clipping function in maintaining stability under comparable circumstances. These findings offer valuable insights for the development of next-generation reinforcement learning algorithms that prioritize stability in dynamic environments.**

***Index Terms*—Reinforcement Learning, PPO, A2C, Stability**

I. Introduction

Advantage Actor-Critic (A2C) [1] and Proximal Policy Optimization (PPO) [2] are Reinforcement Learning (RL) algorithms from stable-baselines3 [3], which use goal orientated learning where the agent learns from the consequence of its actions, instead of being taught what actions to take [4]. RL is the ability for the agent to identify the value of taking each action when in a specific state, from here it can discover the most optimal actions to take [5], [6]. By comparing the reward received that each agent achieves at various entropy coefficient and discount factor values with the goal of identifying if the values of either hyperparameter effect the stability (measure of consistency in the variance between data points) of the agents significantly. They are both tested using the "Lunar Lander" environment from Gymnasium [7].

A. Reseacrh Aim

The Goal of this research is to identify which algorithm is more stable when the test hyperparameter values are changed. The literature review identifies whether modifying the values of either hyperparameter will lead to significant changes in the stability, as well as identifying the mechanisms behind both algorithms that may lead to any variation between the two in terms of stability. From here a hypothesis is made on the information examined in the literature review.

II. Literature Review

A. Related Literature

Previous research has focused on the comparison of A2C and PPO in terms of performance, such as [8] where they identified that when certain settings are controlled and within the same environment and seed, both algorithms output the exact same result. This could support the hypothesis that there is minimal to no difference in the stability of the two algorithms. However, in contrast to the previously discussed paper, the present study does not control the same settings or adjust the algorithm in any substantial way, so the results are likely to have discrepancies. In [9] a comparison between A2C and PPO was made using Multi-Task Learning (MTL) in Atari game environments and found that PPO performed better in 3 of the 4 environments tested.Also compared to the baseline tests A2C was more stable than PPO when MTL was added. This paper compared A2C and PPO with Deep Q Network (DQN) when the learning rate and discount factor is modified, it found that PPO and A2C did better at a higher value for discount factor. With the paper claiming this was especially present in PPO due to "the algorithm's inherent stability". One main issue with this paper is that discount factor was only tested within a very small range, so the results may not be representative. [10]. The main issue with using any of these articles as evidence for a hypothesis is that they did not provide any figures on how stable the results were and because of this minimal insight can be gained directly.

Due to the lack of research on how the stability of either algorithm is affected when the hyperparameters change, this literature review will focus on how each algorithm functions to determine a sensible hypothesis.

B. Policies

Both A2C and PPO use policies [11] that map each state to a single action. A state is a representation of the environment that is observable to the agent at a given time. The action an agent can take is determined by the state in which it is in; some actions can be taken in multiple states, while others are limited and may not be usable in certain states [12]. The agent aims to learn the policies that maximise the reward it will receive, it does this by increasing the probability of choosing actions that lead to more reward while decreasing the probability of choosing actions that lead to less reward [**?**].

$$L_{\text{PG}}(\theta) = \hat{\mathbb{E}}_t \left[\log \pi_\theta(a_t|s_t) \hat{A}_t \right] \quad (1)$$

The previous equation is the logic behind the policy and how it adjusts based on the previous action. Where π_θ is a stochastic policy and $\hat{A}_t$ is an estimator of the advantage function at timestep t [2].

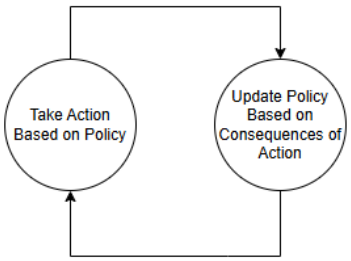

Fig. 1: Policy Diagram Created using Draw.io.

C. Clipping

PPO uses 'clipping' that prevents excessively large policy changes and leads to stable and reliable results, while being easier to implement and suitable for more settings than other RL algorithms [2]. This is crucial when trying to obtain consistent results, as an action that results in a significant change in the returned reward can lead to the policy assigning it a substantially different probability of being selected, even when it does not ensure that the reward value will vary by that much each time, which can lead to a wide range of final reward values [14], [15] which is undesirable for stable results.

$$L_{\text{CLIP}}(\theta) = \hat{\mathbb{E}}_t \left[\min \left(r_t(\theta)\hat{A}_t, \text{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right) \hat{A}_t \right) \right] \quad (2)$$

"Clipping" achieves this in this part of the formula $\text{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right) \hat{A}_t$ which modifies the objective by clipping the probability ratio, which removes the incentive to move probability ratio $r_t(\theta)$ outside of the interval [-1, +1]. Next the formula takes the minimum of both the clipped and unclipped objective, this is done so that the final objective is a pessimistic bound, which prevents any probability changes to the policy if they would make it improve, and includes any which decrease the probability [2].

D. Actor-Critic Algorithms

A2C is an Actor-Critic algorithm, this means it is comprised of two main components, the actor and the critic, the actor uses policies while the critic provides a value estimate or "baseline" that is used by the advantage function [16].

The advantage function that calculates how much more or less rewarding choosing a given action is over following the current policy. This is particularly useful when a given environment uses only positive rewards, as other RL algorithms will never decrease the error probability of taking an action whereas the advantage function uses a comparison to ensure it is taking the more "rewarding" action always [11]. The actor is updated by the policy while the critic is updated using a value method that focuses on decreasing the temporal difference (TD) which is described as the difference between the observed and predicted result [17], [18].

E. Entropy Coefficient

The entropy coefficient (β) can be tuned to increase or decrease exploration (actions that allow the agent to seek out new states and learn more about the environment) [19]. A smaller value can lead to a more deterministic policy while a higher value can lead the policy to be more stochastic [20]. Both A2C and PPO have methods to reduce the likelihood of taking actions that lead to large variances in reward given, however a large β value may lead to this occurring anyway. A lower value can lead to a higher likelihood of reaching local optima and convergence which may be more of a problem for A2C, as it lacks the clipping method used by PPO and instead uses the advantage function along with the critic. The increased stability from clipping reduces the chance of getting stuck in local optima [2] which may lead to PPO returning more stable results than A2C.

F. Discount Factor

The Discount factor (γ) can be tuned to increase or decrease how much the agent prioritises the future reward, with lower value the agent will prioritise actions that lead to immediate rewards while a higher value of γ prioritises actions that lead to future rewards instead [21]. Since time is a factor in this environment the agent may display unexpected behaviour, such as going into an uncontrolled free fall to reach the goal in the quickest time to receive a higher reward, this will be most likely to occur at a higher γ value. There is no significant difference in how either algorithm uses the discount factor [1], [2] that would suggest that the discount factor would affect either differently, however due to the complex nature of RL algorithms there may be a difference that is not obvious without deeper analysis.

G. Learning Rate

The learning rate determines how quickly the model adjusts to an environment. If the learning rate is too low the model converges slowly, and more epochs will be required to achieve an optimal result, whereas a high learning rate may converge to a sub optimal solution [22].

H. Null Hypothesis

The null hypothesis is that there will be no significant difference between the stability of PPO or A2C as γ or β changes.

I. Alternate Hypothesis

The alternative hypothesis is that either PPO is more or as stable as A2C when β is changed and/or that there is a difference between the stability of either algorithm when γ changes. This is due to the "Clipping" function of PPO which

could be superior to the Actor-Critic behaviour of A2C when it comes to reducing the chance of getting stuck in local optima [2] and thus lead to higher stability. There are no mechanisms present in either that would explicitly lead to a difference between the two when γ is modified [1], [2].

There is no expected change in the stability of learning rate between either algorithm due to how having a fixed learning rate per episode functions [22].

III. Methods

A. Experiment Setup

The experiment used the models from stable-baselines3 [2], along with the "Lunar Lander" environment from Gymnasium [7] in order to have an agent and environment. The base code used and expanded upon was from the Falmouth University workshop on AI Gym [23].

With the overall goal of the environment is to land smoothly within the designated landing spot. Within the Lunar Lander environment the agent can take these actions:

- 0: do nothing
- 1: fire left orientation engine
- 2: fire main engine
- 3: fire right orientation engine

The reward the agent receives is:

- increased the closer the lander is to the landing pad.
- increased the faster the lander is moving.
- decreased the more the lander is rotated.

To ensure that the results were accurate 10 episodes were run and the reward achieved logged at 1000 different values for each variable changed, this was done for 5 seeds for a total of 50000 values per hyperparameter per algorithm.

To assess whether the sample size was sufficient to yield reliable results, the following formula for sample size was employed:

$$n = \left(\frac{Z \cdot \sigma}{E} \right)^2 \tag{3}$$

Given the absence of data from prior research, the standard deviation (SD) value (31.49958558) was taken from the results from the experiment that compared A2C and β. While not ideal, this approach provided a rough estimation of the sample size needed. A confidence level of 95% was selected, with a margin of error of ±10. This relatively large margin of error was considered acceptable due to the data being in the hundreds, making the proportional impact of this error smaller. This resulted in an estimated sample size of 38.13, rounded to 39 to decrease the maximum error of the estimate. This study's sample size of 50 exceeds the estimated required sample size by 28.21%, ensuring the validity of the results and supporting that any interpretation of the results is meaningful. The sample size was deliberately kept lower to ensure that the study was completed in a reasonable amount of time while also minimising energy consumption. This consideration reflects the environmental impact of large scale AI training and the carbon footprint such experiments can have [24]. In particular the previous paper highlighted the growing issue of larger training sets and the growing interest in hyperparameter tests, by limiting the sample size the goal is to take an active step to reduce the environmental impact of this research.

The goal was to determine if a change in β or γ affected the stability of either PPO or A2C. Additionally, the learning rate was also modified, this was done to ensure that any results achieved from the tested hyperparameters did not correlate to the reward returned from varying the learning rate. This experiment has had minimal exploration in previous research, so this paper aims to provide insight and evidence as to these claims.

B. Coding the experiment

The code was programmed in PyCharm using python 3.12 as the interpreter. To quickly and easily modify hyperparameters for each algorithm they were set up as arguments in functions. There are 4 functions initialised for this reason, two for each algorithm, one for modifying the hyperparameters and another control that is used for testing the learning rate with standard settings from stable-baselines3.

To test each hyperparameter the code loops through every value of a hyperparameter from 0 to 1 in increments of 0.001, ensuring to run the appropriate RL algorithm in the environment for each value and store the reward returned in a list for data analysis. For every test any control hyperparameter were kept to the same value for all of the experiments, to ensure that fair comparisons can occur.

To ensure that the results could be reproduced, seeding was used. This sets the initial conditions for "randomness" and because of this, the results can be reproduced by setting the seed and hyperparameters to the same values [25].

C. Data analysis and processing

From the acquired data multiple values were calculated, specifically the variance (Var), range, SD, and coefficient of variance (CV). These variables indicate higher stability when they are closer to 0, this is used in comparisons between the experiments to identify the stability of the results relative to one another. For the calculation of A2C vs Learning Rate at a set seed of 50 the result for range was likely an outlier, as it was 196.4% bigger than the next highest value for that experiment. For this reason it was removed from any calculations to ensure the results obtained were accurate.

To account for the limitations of variance metrics with negative values and ensure fair comparisons can occur, the absolute CV values were used instead of the returned results.

To graph this data, the Seaborn library [18] and matplotlib [19] were used to generate visualizations. To enhance accessibility and ensure clarity for all readers, the "colourblind" colour palette was employed for the majority of plots. This approach allows data analysis by those with colour vision deficiencies. To enhance the readability of the graphs the hyperparameter values have been abbreviated to:

- Entropy Coefficient = EC
- Discount Factor = DF
- Learning Rate = LR

The Hex Bit plot from Matplotlib shows the frequency of points that are within a certain range of values. This made it much clearer to identify both the distribution and the density of data points, along with reducing over plotting.

IV. Results

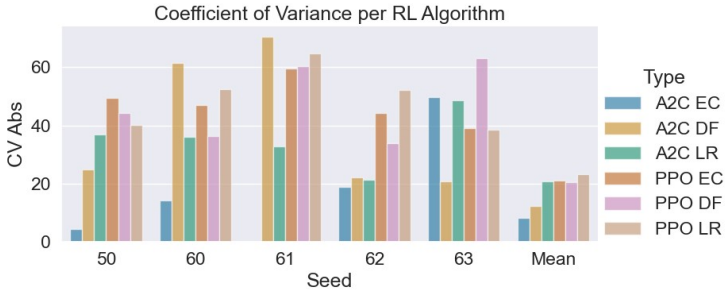

Fig. 2: Coefficient of Variance vs Algorithms

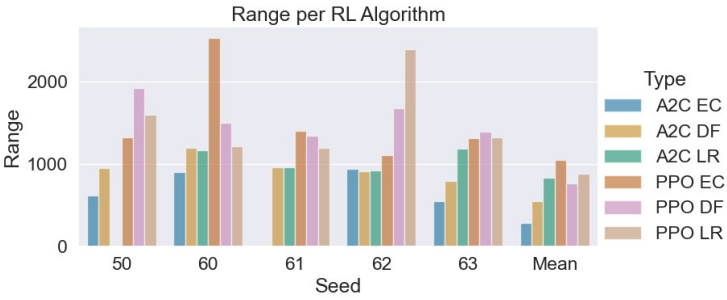

Fig. 3: Range vs Algorithms

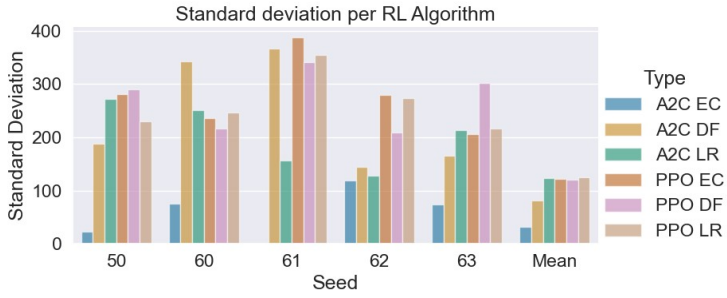

Fig. 4: Standard Deviation vs Algorithms

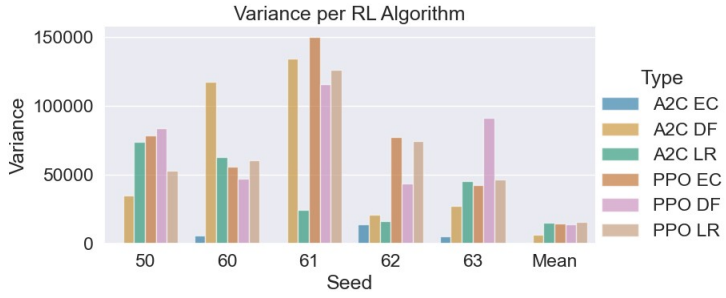

Fig. 5: Variance vs Algorithms

In Fig. 2,3,4,5 the graphs show the CV, Range, SD and Var accordingly for both algorithms and the three hyperparameters being changed. For the purpose of analysis the higher the previous values are, the higher the instability of that test.

The most important value is the CV, this is a standardized measure of the variability of a data set that can be used to compare the stability of multiple data sets even with different scales or units.

A. Entropy Coefficient

Table 1 indicates that the results from A2C EC were all less than 40% of the values obtained by the same test with PPO, indicating that A2C is significantly more stable when EC is changed than it's counterpart. A2C EC exhibited the lowest results in the whole analysis, which proves that it was the most stable test conducted. Contrary to the hypothesis A2C is more stable than PPO when β is modified, which is not supported by the evidence examined in the literature review. This could suggest that there is a mechanism in A2C that leads to a substantial reduction in instability that PPO lacks, this would be a good point for further research into why this occurs and how any findings can be utilised to improve future RL algorithms.

B. Discount Factor

As evidenced by Table 2 the values for A2C DF were less than 73% of the results returned by PPO, indicative of A2C DF being substantially more stable. This was an unexpected result as the hypothesis concluded that there will be no significant difference between then two, this suggests that both models interact differently with γ and that A2C is more successful when optimizing both immediate and future rewards with stability in mind.

C. Intra-Algorithm Analysis

Interestingly Fig. 6 & Table 3 show that A2C DF was significantly more unstable than A2C EC (66%)(Table 3) whereas PPO DF and PPO EC showed less of a change with only the CV (34%) and Range (73%) being significantly different , suggesting that A2C suffers more proportionally when DF is changed compared to PPO. Additionally PPO is slightly more susceptible to instability when β changes than γ, future research could focus on trying to bridge this gap to improve the stability of the whole model.

D. Learning Rate

The results from A2C and PPO when LR is changed were similiar and there was not a significant difference between the returns of either across all the tests. This could suggest that the mechanisms behind either algorithm have no bearing on how the learning rate effects stability, this is to be expected and helps to confirm that the results are accurate.

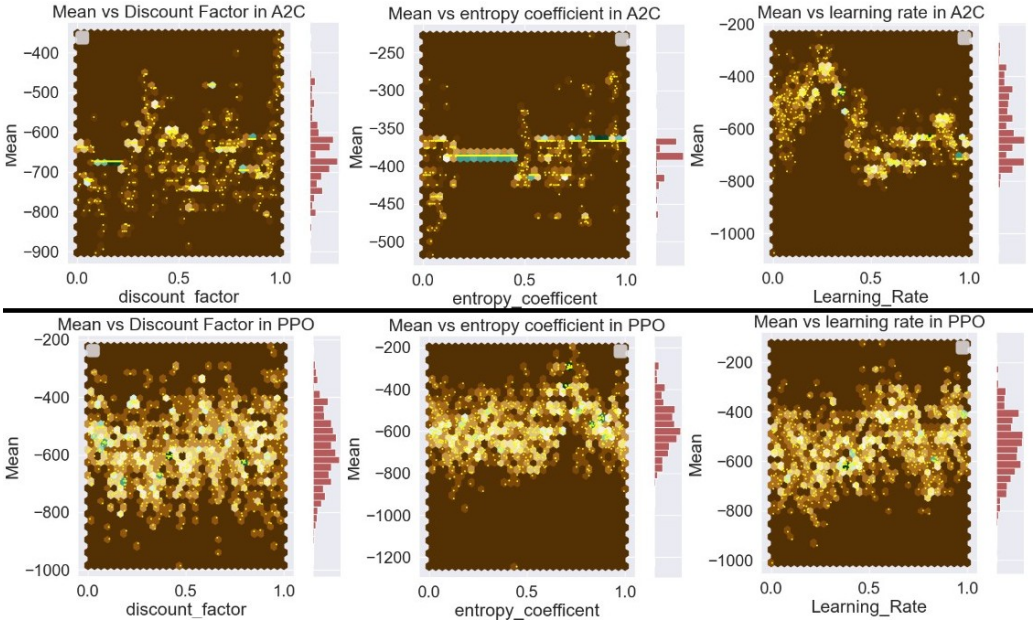

Fig. 6: Mean Reward of 5 seeds for each Algorithm

Value	A2C EC	PPO EC	% of PPO EC
Variance	991.2327	14870.42	6.67%
Standard Deviation	31.49959	74.19%	25.81%
Coefficient of Variance	8.14446	61.61%	38.39%
Range	289.9932	1044.523	27.77%

TABLE I: Comparison between A2C EC and PPO EC

Value	A2C DF	PPO DF	% of PPO DF
Variance	6559.216	14365.21	45.67%
Standard Deviation	81.02947	119.9149	67.56%
Coefficient of Variance	12.36426	20.61335	60.00%
Range	550.1918	760.1694	72.35%

TABLE II: Comparison between A2C DF and PPO DF

Value	A2C EC	A2C DF	% of A2C DF
Variance	991.2327	6559.216	15.11%
Standard Deviation	31.49959	81.02947	38.87%
Coefficient of Variance	8.14446	12.36426	65.87%
Range	289.9932	550.1918	52.68%

TABLE III: Comparison between A2C EC and A2C DF

Value	PPO EC	PPO DF	% of PPO EC
Variance	14870.42	14365.21	96.60%
Standard Deviation	74.19	119.9149	161.64%
Coefficient of Variance	61.61	20.61335	33.45&
Range	1044.523	760.1694	72.78%

TABLE IV: Comparison between PPO EC and PPO DF

V. Conclusion

This paper addressed the proposed question and identified that varying both the entropy coefficient and the discount factor had a significant impact on the stability of both A2C and PPO. A2C demonstrated notably greater stability when the entropy coefficient was tested compared to when the discount factor was altered.

Contrary to the initial hypothesis, the results showed that A2C was more stable than PPO regardless of which hyperparameter was tested. This outcome could be attributed to several factors. For example, the Actor-Critic mechanism may be more effective at regulating results and mitigating instability than initially assumed. Alternatively, the use of multiple workers in A2C might influence its stability differently when compared with PPO.

Limited research exists on how RL algorithms respond to changes in hyperparameters, particularly in terms of stability. It is hoped that this paper provides valuable context and insight into how RL algorithms behave under these conditions, enabling future research to make more informed decisions when selecting RL algorithms based on performance to consider the stability of a system with more import.

The findings of this study highlight the importance of considering the stability of RL models to ensure consistent and reliable results. Additionally, they could inspire efforts to enhance PPO and future RL algorithms by drawing on the mechanisms identified in this paper that may lead to the notable stability of A2C.

However, this paper did not examine all of the hyperparameters or a broad spectrum of RL algorithms. Future work could benefit from conducting more extensive experiments that include a comprehensive comparison of popular RL models. Moreover, limited analysis was conducted on the code of the tested algorithms, which might help uncover the underlying causes of the observed results.

Future research should focus on understanding why specific RL models exhibit greater stability and on designing new models with stability as a key objective. Such advancements

could have critical applications in fields like medicine for instance, developing a highly stable model capable of accurately identifying and diagnosing cancer cells within specimens could potentially save thousands of lives while minimising the change of a false positive.

By analyzing the stability of both A2C and PPO, we can glean a deeper understanding of when and why we should use RL models and how these models need to be tailored to their environments for optimal results. A2C demonstrates significantly more stability than PPO when both entropy coefficient and discount factor are tested. However, this should not lead to the outright dismissal of PPO, as the strength and weakness of RL models lie in their varying suitability to specific environments. Ultimately, the knowledge required to determine the best tool for a given problem will avail future researchers far more than any single empirical analysis such as this ever could.

A. Legal and ethical consideration

The tests explored within this paper were designed to ensure that a minimal amount of misuse could occur with this research, however when discussing future uses for similar research it must be acknowledged that the applications must be carefully examined before being put into practice. For example if this idea of an extremely stable algorithm was used in a medical setting it is possible due to the "black-box" like nature of RL models that a mistake in the code or any tampering could cause incorrect diagnoses. This may cause substantial harm as the RL model was believed to be extremely reliable and as such it's judgments weren't given enough scrutiny.

Beyond the medical field, these concerns could extend to financial systems, where errors might result in severe economic repercussions, or to autonomous vehicles, where safety depends heavily on error-free algorithms that respond "correctly" every time. There can be no room for mistakes when human life is on the line.

All sources of code or knowledge should be cited and accredited appropriately.

VI. References

References

[1] V. Mnih et al., "Asynchronous Methods for Deep Reinforcement Learning," arXiv, Jun. 16, 2016. [Online]. Available: https://doi.org/10.48550/arXiv.1602.01783

[2] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, 'Proximal Policy Optimization Algorithms', Aug. 28, 2017, arXiv: arXiv:1707.06347. doi: 10.48550/arXiv.1707.06347.

[3] 'Base RL Class — Stable Baselines3 2.6.0a1 documentation'. Accessed: Feb. 17, 2025. [Online]. Available: https://stable-baselines3.readthedocs.io/en/master/modules/base.html

[4] S. Ravichandiran, *Hands-On Reinforcement Learning with Python: Master Reinforcement and Deep Reinforcement Learning Using OpenAI Gym and TensorFlow*, 1st ed. Packt Publishing, Limited, 2018.

[5] 'Systematic Review of Reinforcement Learning Literature'. Accessed: Feb. 15, 2025. [Online]. Available: https://futuremachinelearning.org/systematic-review-of-reinforcement-learning-literature/?form=MG0AV3

[6] [1] R. S. Sutton and A. Barto, Reinforcement learning: an introduction, Nachdruck. in Adaptive computation and machine learning. Cambridge, Massachusetts: The MIT Press, 2014.

[7] 'Lunar Lander - Gymnasium Documentation'. Accessed: Feb. 17, 2025. [Online]. Available: https://gymnasium.farama.org/environments/box2d/lunar_lander/

[8] S. Huang, A. Kanervisto, A. Raffin, W. Wang, S. Ontañón, and R. F. J. Dossa, 'A2C is a special case of PPO', May 18, 2022, arXiv: arXiv:2205.09123. doi: 10.48550/arXiv.2205.09123.

[9] C. Nalty, "A Comparison of Policy Gradient Methods for Multitask Learning," [Online]. Available: https://api.semanticscholar.org/CorpusID:245851528

[10] N. D. L. Fuente and D. A. V. Guerra, 'A Comparative Study of Deep Reinforcement Learning Models: DQN vs PPO vs A2C', Jul. 19, 2024, arXiv: arXiv:2407.14151. doi: 10.48550/arXiv.2407.14151.

[11] 'Reinforcement Learning with Policy Iteration | by Amit Yadav | Biased-Algorithms | Medium'. Accessed: Feb. 16, 2025. [Online]. Available: https://medium.com/biased-algorithms/reinforcement-learning-with-policy-iteration-52b0c3bc8c06

[12] 'Fundamentals of Reinforcement Learning: Policies, Value Functions, & Bellman Equations'. Accessed: Feb. 16, 2025. [Online]. Available: https://blog.mlq.ai/reinforcement-learning-policies-value-functions-bellman-equation/

[13] 'Understanding PPO: A Game-Changer in AI Decision-Making Explained for RL Newcomers | by Chris Hughes | Medium'. Accessed: Feb. 17, 2025. [Online]. Available: https://medium.com/@chris.p.hughes10/understanding-ppo-a-game-changer-in-ai-decision-making-explained-for-rl-newcomers-913a0bc98d2b

[14] DeepSeek-R1 Dissection: Understanding PPO & GRPO Without Any Prior Reinforcement Learning Knowledge | Yihua's Blog'. Accessed: Feb. 17, 2025. [Online]. Available: https://normaluhr.github.io/2025/02/07/grpo/

[15] 'Proximal Policy Optimization — Spinning Up documentation'. Accessed: Mar. 12, 2025. [Online]. Available: https://spinningup.openai.com/en/latest/algorithms/ppo.html

[16] 'Actor-Critic Methods, Advantage Actor-Critic (A2C) and Generalized Advantage Estimation (GAE) - Alexander Van de Kleut'. Accessed: Feb. 17, 2025. [Online]. Available: https://avandekleut.github.io/a2c/

[17] 'Advantage Actor Critic Tutorial: minA2C | Towards Data Science'. Accessed: Mar. 12, 2025. [Online]. Available: https://towardsdatascience.com/advantage-actor-critic-tutorial-mina2c-7a3249962fc8/

[18] 'Advantage Actor-Critic (A2C) Algorithm Explained and Implemented in PyTorch | by Dixitaniket | Medium'. Accessed: Mar. 12, 2025. [Online]. Available: https://medium.com/@dixitaniket76/advantage-actor-critic-a2c-algorithm-explained-and-implemented-in-pytorch-dc3354b60b50

[19] 'What is the fundamental difference between exploration and exploitation in the context of reinforcement learning? - EITCA Academy'. Accessed: Feb. 17, 2025. [Online]. Available: https://eitca.org/artificial-intelligence/eitc-ai-arl-advanced-reinforcement-learning/tradeoff-between-exploration-and-exploitation/exploration-and-exploitation/examination-review-exploration-and-exploitation/what-is-the-fundamental-difference-between-exploration-and-exploitation-in-the-context-of-reinforcement-learning/

[20] J. Liu, X. Gu, and S. Liu, 'Policy Optimization Reinforcement Learning with Entropy Regularization', Oct. 16, 2020, arXiv: arXiv:1912.01557. doi: 10.48550/arXiv.1912.01557.

[21] G. Paczolay and I. Harmati, 'NPV-DQN: Improving Value-based Reinforcement Learning, by Variable Discount Factor, with Control Applications', ACTA POLYTECH HUNG, vol. 21, no. 11, pp. 175–190, 2024, doi: 10.12700/APH.21.11.2024.11.10.

[22] 'Learning Rate in Neural Network - GeeksforGeeks'. Accessed: Mar. 12, 2025. [Online]. Available: https://www.geeksforgeeks.org/impact-of-learning-rate-on-a-model/

[23] 'Using Python for Machine Learning | COMP213'. Accessed: Feb. 25, 2025. [Online]. Available: http://fal.fosslab.uk/comp250/ml/08_learning/

[24] G. Tamburrini, 'The AI Carbon Footprint and Responsibilities of AI Scientists', Philosophies, vol. 7, no. 1, p. 4, Jan. 2022, doi: 10.3390/philosophies7010004.

[25] 'Understanding Seeds in AI: The Key to Reproducibility and Creativity | by Nikunj Vaghasiya | Medium'. Accessed: Feb. 26, 2025. [Online]. Available:

VII. APPENDICE

A. Workers

Workers in RL algorithms explore the environment in parallel with the agent, with the objective of increasing the effectiveness of the policy. By taking actions and observing the outcomes, the workers update the policy similarly to the agent itself, these updates occur in sync with the agent as it explores the environment leading to a more effective policy over time when compared with an agent that doesn't use workers. This paper observed that the multiple workers of A2C increased stability [1], however, PPO also uses multiple workers and as such the increased stability may not lead to a substantial difference between the two.

TABLE V: Comparison of A2C and PPO Experiments

Measure	**A2C EC**	**A2C DF**	**PPO EC**	**PPO DF**
Variance	991.2327	6559.216	14870.42	14365.21
Standard Deviation	31.49959	81.02947	74.19	119.9149
Coefficient of Variance	8.14446	12.36426	61.61	20.61335
Range	289.9932	550.1918	1044.523	760.1694

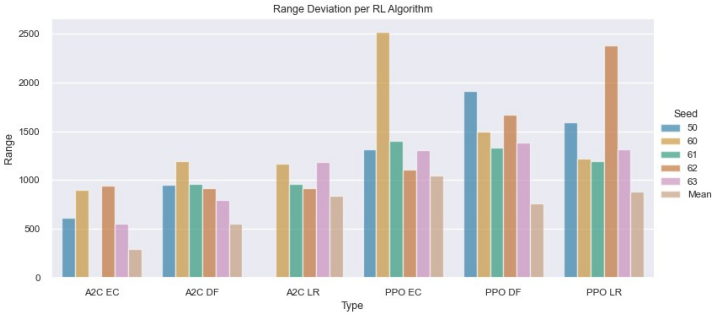

Fig. 7: Range vs Algorithms

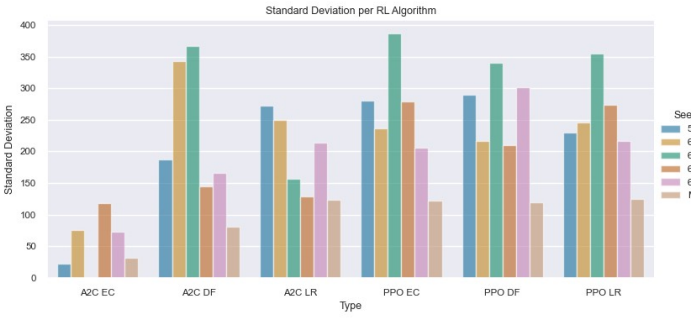

Fig. 8: Standard Deviation vs Algorithms

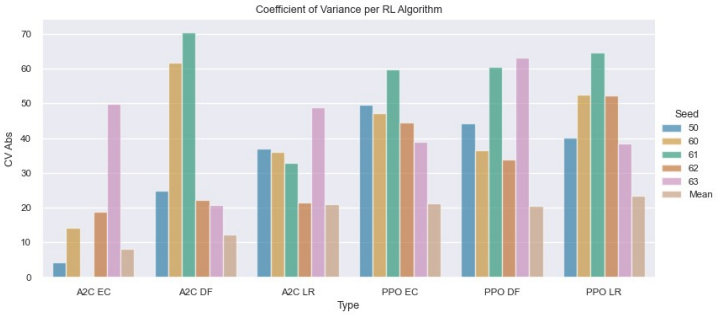

Fig. 9: Coefficient of variance vs Algorithms

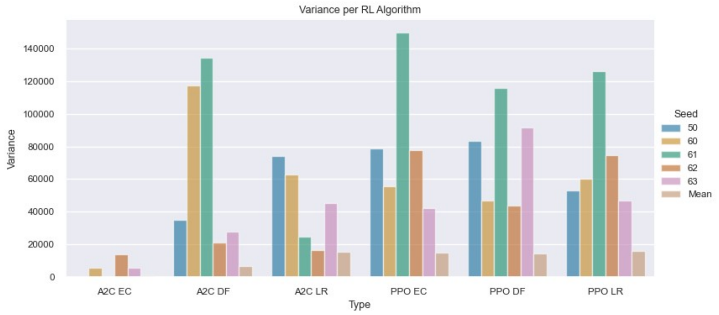

Fig. 10: Variance vs Algorithms