Most Compatible Reinforcement Learning Algorithm for Deep Brain Stimulation

Ian Paik Choe

Abstract

Tremors are a symptom of Parkinson's disease that causes involuntary shaking movements in the hands and other parts of the body, which can disturb one's quality of life. Tremors happen when malfunctioning neurons synchronize. Therefore, suppression and control of this collective synchronous activity are of great importance. Deep brain stimulation is where surgeons decide the amplitude and frequency of the stimulation to nullify collected signals from synchronous neurons in the brain according to their observation and expertise. A virtual Reinforcement Learning environment Krylov et al. created in 2020 can simulate this collective behavior of neurons and allows us to find suppression parameters for the environment of synthetic degenerate models of neurons. Although the newest generation of Deep Brain Stimulation technology does provide feedback functionality (they can be controlled using both traditional, physical controllers and machine algorithms), it is still challenging to decide which algorithm is most suitable for the task; the study by Krylov et al. applies Proximal Policy Optimization to their environment and successfully suppresses the synchronization in neuron activity. However, they do not test other types of algorithms. This paper expands upon their findings by systematically evaluating six reinforcement learning algorithms (A2C, DDPG, PPO, SAC, TD3, and TRPO). Our results indicate that Trust Region Policy Optimization (TRPO) is the most effective under conditions of low learning rate, moderate divergence between updates, and prioritizing long-term rewards.

I. Introduction

Parkinson's disease is a prevalent neurodegenerative disorder affecting approximately one million individuals in the United States, and approximately 60,000 new cases are diagnosed annually [1]. One primary symptom of Parkinson's disease is tremors, which cause involuntary shaking in various body regions, notably the hands and other limbs [2]. These tremors can significantly disrupt an individual's overall quality of life and prevent them from performing tasks requiring precision. Tremors are caused by malfunctioning neurons firing in synchronization [3].

Among the treatments available (such as medication, surgery, and therapy), deep brain stimulation has emerged as a promising treatment for the tremors associated with Parkinson's disease [4]. Deep brain stimulation involves a surgical procedure in which an electrical pulse is administered to a specific brain region through a pacemaker to nullify the synchronization of erroneous brain signals to diminish unwanted movements. Surgeons play a pivotal role in determining the stimulation parameters, processing both the amplitude and frequency of the synchronized signals, and returning a counter-pulse that matches the amplitude and frequency, drawing upon their clinical observations and professional expertise to make these calculations [5].

In recent years, the evolution of deep brain stimulation technology has incorporated feedback mechanisms into the pacemakers, which can be controlled using traditional methods (manually) or machine learning algorithms [6]. While these advancements are a significant stride forward in the field, the challenge of selecting optimal algorithms for guiding deep brain stimulation persists. The complexity of the neural dynamics involved necessitates careful consideration of algorithmic choices to ensure the efficacy of stimulation strategies [7].

Prior research has illuminated potential avenues for addressing the intricate issue of synchrony within brain neurons. Notably, the original study, which created the gym environment (a virtual environment that simulates a situation for reinforcement learning) for deep brain stimulation, demonstrated the efficacy of the algorithm Proximal Policy Optimization (PPO) in mitigating tremors [1]. This foundational work not only underscores the relevance of algorithmic innovation in the context of deep brain stimulation but also sets the stage for further exploration and refinement of the area.

As the prevalence of Parkinson's disease continues to rise [9] and the pursuit of enhanced therapeutic strategies gains momentum, it becomes imperative to examine the algorithms conducting deep brain stimulation. This study aims to contribute to this issue by investigating potential reinforcement learning algorithms and under which hyperparameters these algorithms work optimally. Through empirical investigation and theoretical analysis, this study seeks to extend the accuracy of deep brain stimulation through reinforcement learning algorithms, thereby paving the way for better therapeutic interventions and improved quality of life for Parkinson's disease patients.

This paper will assess the efficacy of six distinct reinforcement learning algorithms – PPO, Deep Deterministic Policy Gradient (DDPG), Trust Region Policy Optimization (TRPO), Soft Actor-Critic (SAC), Advantage Actor-Critic (A2C), and Twin-delayed Deep Deterministic Policy Gradient (TD3) – within the deep brain stimulation gym environment. Furthermore, this study will identify hyperparameter configurations that demonstrate optimal efficiency within each given environment, employing ablation studies to substantiate the findings. The experiment in this study utilizes an existing gym environment that simulates the treatment and response of deep brain stimulation. It also utilizes the initial studies of the six algorithms being tested. The ablation studies of each algorithm are performed using Optuna, a hyperparameter optimization framework.

II. Related Materials:

1. Reinforcement Learning on Deep Brain Stimulation

Reinforcement Learning is a type of machine learning where an agent learns to make decisions by interacting with an environment to maximize cumulative rewards. The agent takes actions, receives feedback in the form of rewards or penalties, and updates its strategy based on past experiences using trial and error. In simpler terms, the agent continuously improves its behavior by learning from past mistakes and successes, much like how a person refines their skills through practice.

In a study regarding the usage of reinforcement learning in deep brain stimulation by Gao et al. (2020), a patient-specific approach to deep brain stimulation control is introduced using deep reinforcement learning [10]. The basal ganglia region, implicated in Parkinson's Disease, is modeled as a Markov decision process. A Markov decision process (MDP) is a mathematical framework used to describe decision-making where outcomes depend on both current actions and probabilistic future states. The Basal Ganglia neuron activity defined the state space, and the action space consisted of stimulation patterns. The study aimed to maximize cumulative rewards over a treatment duration while limiting the stimulation frequency. This means that the system sought to optimize the effectiveness of the treatment while minimizing unnecessary electrical stimulation, which can reduce side effects and energy consumption. Using a Brain-on-Chip FPGA platform implementing a physiologically relevant Basal Ganglia model, the reinforcement learning-based deep brain stimulation controllers demonstrate superior energy efficiency compared to fixed-frequency controllers. Traditional DBS methods deliver constant electrical stimulation, but this reinforcement learning approach allows the system to adapt and optimize stimulation in real-time based on the patient's specific neural activity. This approach reduced energy consumption and effectively mitigated Parkinson's Disease symptoms, highlighting its potential clinical significance.

Krylov et al. (2020) leveraged modern machine-learning techniques to address self-sustained collective oscillations observed in degenerated neurons of patients with Parkinson's disease [11]. The proposed hybrid model combined an oscillator environment with a policy-based reinforcement learning component. Model-agnostic synchrony control was achieved through proximal policy optimization and artificial neural networks in an Actor-Critic configuration. The successful demonstration of synchrony suppression for diverse neuronal ensembles underscores the versatility of this approach in addressing pathological signaling.

An additional study by Krylov et al. (2020) presents the gym environment used in this study as a platform for investigating synchronization suppression without the ethical and practical constraints associated with live human experimentation [8]. The framework's effectiveness and stability are shown through successfully suppressing synchrony neurons (which causes brain tremors). Moreover, integrating multiple Proximal Policy Optimization (PPO) agents further enhances the framework's ability to eliminate unwanted oscillations.

2. Algorithms being tested

i. Proximal Policy Optimization (PPO) [13]

PPO improves upon the TRPO algorithm by enhancing its sample efficiency. It operates by iteratively engaging with the environment and refining a surrogate objective function using stochastic gradient ascent. Unlike conventional approaches that update based on individual data samples, PPO introduces a novel objective function that facilitates multiple epochs of minibatch updates. In short PPO efficiently fine-tunes the agent's actions while preventing drastic changes that could destabilize the system.

ii. Deep Deterministic Policy Gradient (DDPG) [14]

DDPG is a type of reinforcement learning algorithm that follows the actor-critic method and does not rely on models of the environment. It is particularly useful in situations where actions can take on continuous values rather than being limited to discrete choices. The main advantage of DDPG is its ability to learn effective decision-making strategies that match or even exceed the performance of methods that have full knowledge of the environment's dynamics. Additionally, it can learn directly from raw image data (such as camera feeds) without requiring pre-processed information.

iii. Trust Region Policy Optimization (TRPO) [15]

TRPO is an advanced reinforcement learning algorithm designed to improve decision-making in complex scenarios. It builds on the idea of "natural policy gradients," which optimize decision-making strategies more effectively than traditional gradient-based methods. TRPO is particularly useful for training deep neural networks to make decisions, as it ensures that updates to the decision-making policy are both stable and gradual. This helps the algorithm improve steadily over time without requiring extensive tuning of parameters.

iv. Soft Actor Critic (SAC) [16]

SAC is a deep reinforcement learning algorithm that follows the actor-critic approach but incorporates the concept of "maximum entropy reinforcement learning." This means that SAC not only tries to maximize rewards but also encourages some level of randomness in decision-making, which helps the agent explore different possibilities more effectively. Unlike previous approaches that relied on Q-learning, SAC combines off-policy learning (learning from past experiences rather than just recent ones) with a stable actor-critic structure. As a result, it consistently outperforms other methods in tasks requiring continuous control and demonstrates robustness even when trained under different random conditions.

v. Advantage Actor Critic (A2C) [17]

A2C is another actor-critic-based reinforcement learning algorithm that optimizes decision-making using a method called asynchronous gradient descent. This means that the algorithm can update its learning process using multiple calculations happening at the same time, which speeds up training. A2C has proven to be very efficient, achieving better performance in playing Atari games while using only half the training time compared to other leading approaches. Additionally, A2C performs well in various motor control tasks and can even navigate 3D environments based on visual inputs.

vi. Twin-delayed Deep Deterministic Policy Gradient (TD3) [18]

The TD3 algorithm is a value-based reinforcement learning approach that addresses the issue of overestimated value estimates and suboptimal policies. It extends the concept of Double Q-learning by introducing mechanisms to mitigate overestimation bias in both the actor and critic components of an actor-critic architecture. This is achieved by utilizing a pair of critics and taking the minimum value between them to constrain overestimation. The algorithm establishes a correlation between overestimation bias and target networks, and proposes a strategy of delaying policy updates to reduce the error introduced per update, thereby enhancing overall performance.

3. Optuna for Hyperparameter Tuning [19]

Optuna is a software tool designed to automate and improve the process of optimizing hyperparameters, which are key settings that influence the performance of machine learning models. Unlike traditional methods that use predefined search spaces, Optuna follows a flexible "define-by-run" approach. This means that instead of specifying all possible settings in advance, users can dynamically construct and adjust the search space as the optimization process progresses. This results in more efficient searches and better tuning of machine learning models.

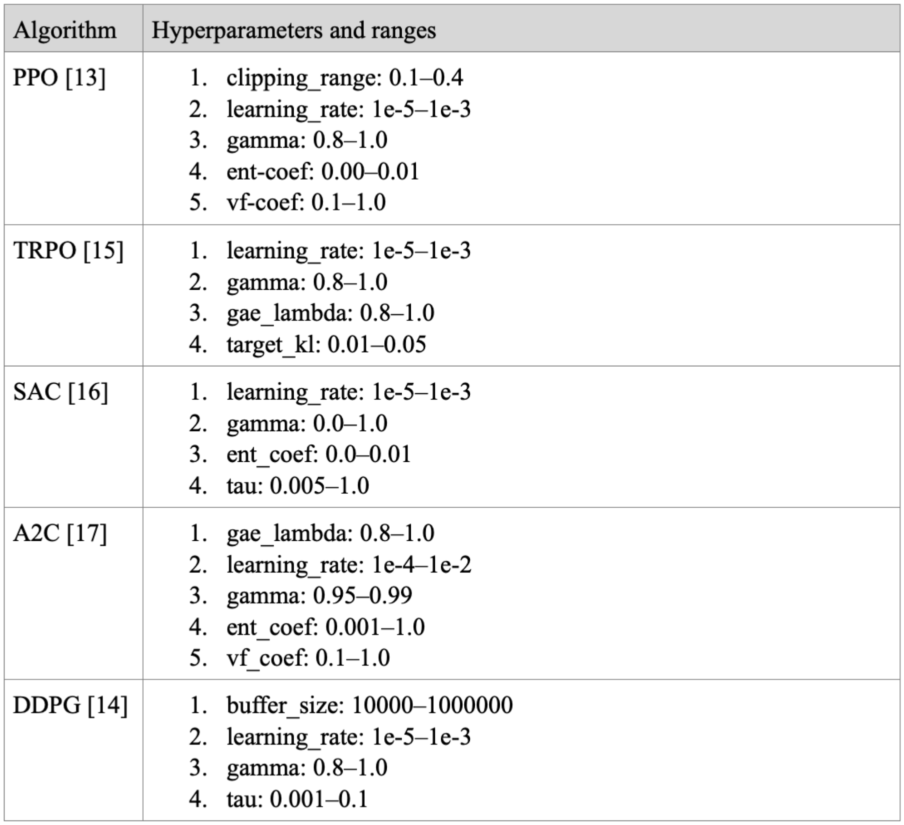
One of Optuna’s strengths is its ability to implement effective search and pruning strategies, ensuring that computational resources are used efficiently. By intelligently stopping unpromising trials early, Optuna speeds up the optimization process while maintaining high accuracy. The development of Optuna required careful design and empirical testing to ensure its effectiveness. Real-world applications and experimental results demonstrate how this novel approach enhances model performance compared to conventional hyperparameter tuning methods.

III. Methods

In this section, we outline the methodology employed for finding the ultimate reinforcement learning model. The purpose of this study is to find the most suitable reinforcement learning algorithm for the Deep Brain Stimulation gym environment, as well as maximize the performance of the model through a series of ablation studies to find the best combination of hyperparameters for each algorithm.

1. Ablation Studies and Hyperparameter Tuning:

Before running each Reinforcement Learning algorithm, we conducted a series of ablation studies. The purpose of these studies is to analyze the impact of individual components and hyperparameters on each algorithm's performance. To ensure fair and unbiased comparisons, we used a series of suggested values generated using the Optuna package.



Algorithm	Hyperparameters and ranges
PPO [13]	1. clipping_range: 0.1–0.4 2. learning_rate: 1e-5–1e-3 3. gamma: 0.8–1.0 4. ent-coef: 0.00–0.01 5. vf-coef: 0.1–1.0
TRPO [15]	1. learning_rate: 1e-5–1e-3 2. gamma: 0.8–1.0 3. gae_lambda: 0.8–1.0 4. target_kl: 0.01–0.05
SAC [16]	1. learning_rate: 1e-5–1e-3 2. gamma: 0.0–1.0 3. ent_coef: 0.0–0.01 4. tau: 0.005–1.0
A2C [17]	1. gae_lambda: 0.8–1.0 2. learning_rate: 1e-4–1e-2 3. gamma: 0.95–0.99 4. ent_coef: 0.001–1.0 5. vf_coef: 0.1–1.0
DDPG [14]	1. buffer_size: 10000–1000000 2. learning_rate: 1e-5–1e-3 3. gamma: 0.8–1.0 4. tau: 0.001–0.1

i. Establishing ranges for testing hyperparameters.

Figure 1 is a table which shows each hyperparameter we adjusted for each algorithm, as well as the ranges of those values we are testing. These values are passed through Optuna which finds the optimal combination of values for each hyperparameter. Though these hyperparameters can have a greater range of values, I chose to use a recommended range of values for each due to time constraints and technological limitations.

ii. Average Episode Reward

The Average Episode Reward (AER) is the reward metric for the models tested in this experiment; this is how we determine which combination of hyperparameters to use, as well as which reinforcement learning algorithm is the most successful. The reward prediction is the difference between an output generated by the model and real feedback data. The AER is the average of ten such reward predictions made by the current state of the model.

iii. Testing until convergence

For ten trials per algorithm, using the hyperparameters recommended by Optuna, we tested each algorithm until they reached convergence. When the improvement between the AER of each *step* (2048 timesteps) was both positive and less than eight, we concluded that the model had reached convergence and ended the trial.

iv. Recording the trials using Wandb

For each trial, we used the machine learning development platform Wandb to record the AER after every *step*. The records made in Wandb are characterized by each algorithm which have ten trials each.

v. Recording the best Hyperparameters.

Of the models from the ten trials, we recorded the hyperparameters of the model which converged with the highest AER to a text file document. The optimal set is printed in the console after all the trials by Optuna.

2. Final Evaluation

i. Training the final model

With the recorded optimal hyperparameters for each algorithm, we trained each model for 1,000,000 timesteps.

ii. Recording data using Wandb

In the final evaluation, each *step* is considered to be 1000 timesteps. We, again, used the machine learning development platform Wandb to record the AER every *step* of the final evaluation (1000 *steps* recorded total).

iii. Saving the final model

After training the final models for each algorithm, they are saved to a file.

iv. Performance comparison

Because the merits of each algorithm may lie in different areas, we decided that the reward performance of each algorithm could be evaluated using multiple factors:

1. **Final AER** - the AER when the model is on the 1000s step (end of training). The efficacy of the model can be represented using the final AER because it is the performance of the model once it has finished training.
2. **Peak AER** - the maximum AER reached throughout the training of the model. Sometimes, the final AER may not correctly represent the efficacy model. For instance, if the model ends training on a trough, the final AER underrepresents the overall strength of the model.
3. **Oscillation** - the degree of fluctuation in AER once the model has reached convergence. A low oscillation following convergence represents consistency in the algorithm; this means that outputs from the model are more stable and less likely to vary.
4. **Trough frequency** - the frequency of sudden drops in AER once the model had converged. A low trough frequency means that the algorithm will have less sudden drops in performance.

IV. Results

Evaluating different reinforcement learning algorithms—PPO, TRPO, DDPG, SAC, A2C, TD3—in a gym environment simulating deep brain stimulation yielded valuable insights into their respective performances. The measurement of reward performance was conducted using the aforementioned AER metric.

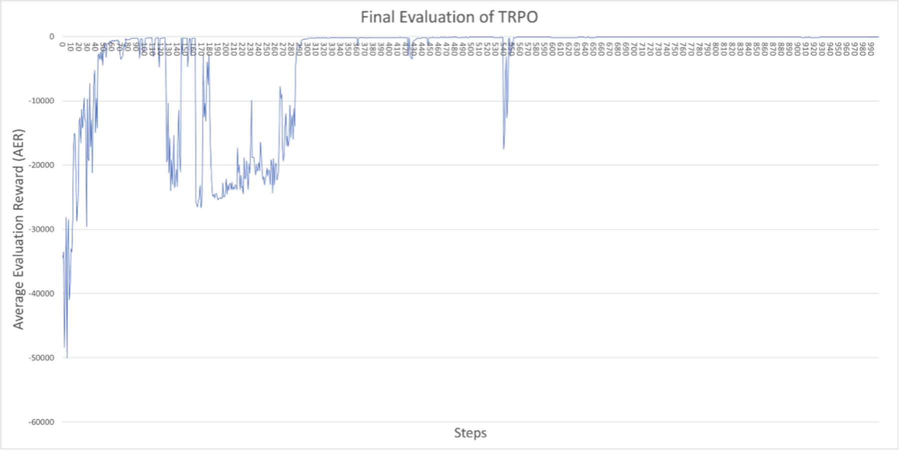


Figure 2

Figure 2 shows the final evaluation of the most successful reinforcement learning algorithm observed, TRPO. Each step (the x-axis of the graph) is equivalent to 1000 timesteps. The figure displays the average evaluation reward (AER) per step throughout the training of TRPO. Among the algorithms tested, TRPO emerged as the most successful algorithm. As defined in the Methods section, each algorithm is assessed based on five criteria: final AER, peak AER, oscillation after convergence, and trough frequency.

TRPO achieved a peak AER of -87.552, second to the peak of TD3. TRPO concluded its evaluation with an AER of -96.199, the greatest final AER. Furthermore, TRPO notably displayed the least troughs in the AER curve, with no observed troughs following *step* 550. All the other algorithms were largely unstable until the end of their training.

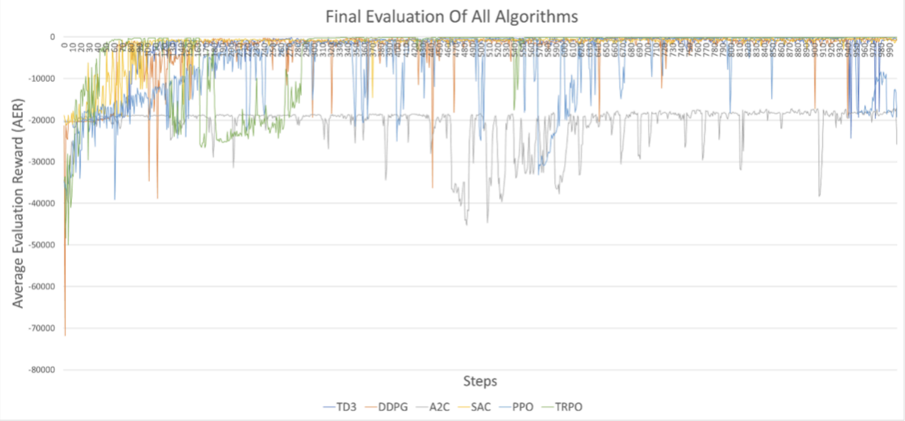
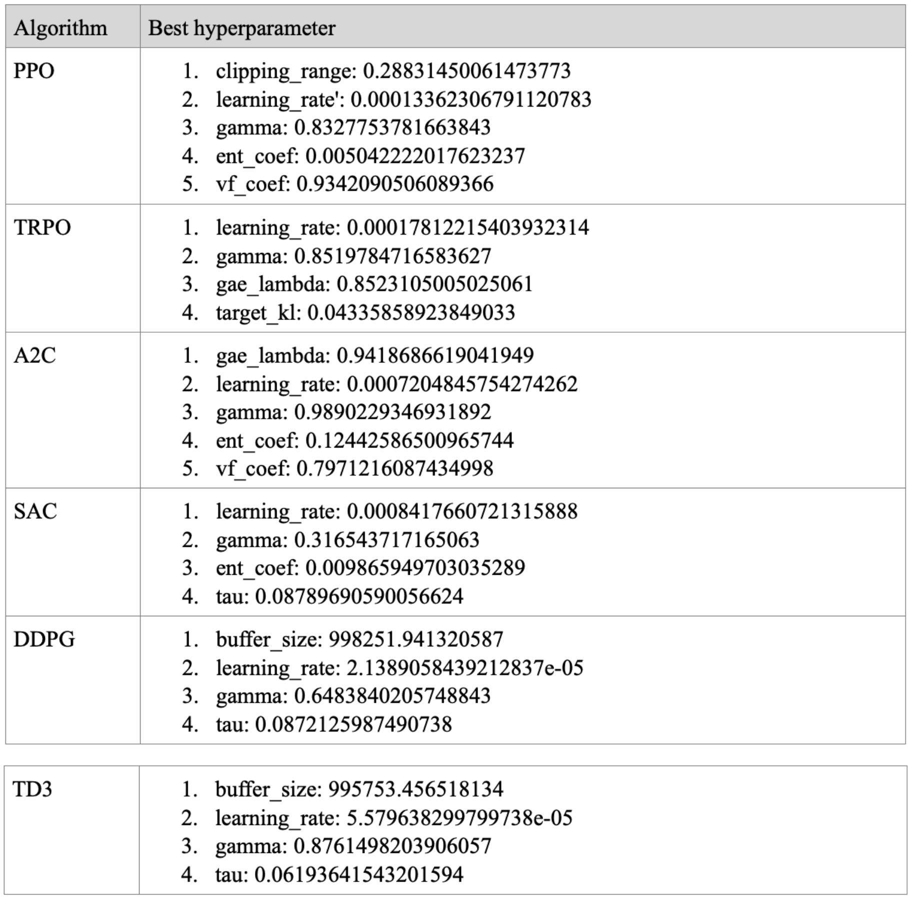


Figure 3



Algorithm	Best hyperparameter
PPO	1. clipping_range: 0.28831450061473773 2. learning_rate': 0.00013362306791120783 3. gamma: 0.8327753781663843 4. ent_coef: 0.005042222017623237 5. vf_coef: 0.9342090506089366
TRPO	1. learning_rate: 0.00017812215403932314 2. gamma: 0.8519784716583627 3. gae_lambda: 0.8523105005025061 4. target_kl: 0.04335858923849033
A2C	1. gae_lambda: 0.9418686619041949 2. learning_rate: 0.0007204845754274262 3. gamma: 0.9890229346931892 4. ent_coef: 0.12442586500965744 5. vf_coef: 0.7971216087434998
SAC	1. learning_rate: 0.0008417660721315888 2. gamma: 0.316543717165063 3. ent_coef: 0.009865949703035289 4. tau: 0.08789690590056624
DDPG	1. buffer_size: 998251.941320587 2. learning_rate: 2.1389058439212837e-05 3. gamma: 0.6483840205748843 4. tau: 0.0872125987490738
TD3	1. buffer_size: 995753.456518134 2. learning_rate: 5.579638299799738e-05 3. gamma: 0.8761498203906057 4. tau: 0.06193641543201594

Figure 4

Figure 3 shows the reward performance, measured in AER, of the final evaluation of each model. Figure 4 shows the optimal set of hyperparameters for each algorithm.

PPO, while displaying a slightly different trajectory from TRPO, also showcased promising results. It reached a peak AER of -152.937 and ended at -19225.487 (which corresponded to a trough in the AER graph). However, consistent troughs accompanied PPO's convergence, and the algorithm required a relatively longer time to reach its final state than TRPO.

Conversely, SAC achieved a peak AER of -256.044 and concluded with an AER of -373.807. Although SAC demonstrated higher stability than PPO, its AER graph displayed oscillations after convergence, indicating variations in performance.

A2C demonstrated unique behavior in its performance profile. It peaked at an AER of -17211.231 and concluded at an AER of -25841.174. A2C exhibited early convergence at approximately 20 steps with an AER of -30000, yet it remained susceptible to frequent oscillations and experienced multiple troughs.

DDPG reached a peak AER of -82.877 and concluded with an AER of -808.598. The algorithm consistently encountered troughs, but exhibited moderate oscillations throughout the evaluation.

TD3 showcased a unique pattern in its performance. It achieved a peak AER of -53.024 and ended with an AER of -637.578. TD3 exhibited fewer troughs but higher oscillations compared to other algorithms.

In summary, comparing these reinforcement learning algorithms on the simulated deep brain stimulation gym environment revealed that TRPO was the most successful in achieving high AER values and stability. PPO demonstrated competitive performance but with longer convergence times and consistent troughs. SAC exhibited stability but with post-convergence oscillations. A2C displayed early convergence but struggled with oscillations. DDPG maintained a moderate performance with frequent troughs, while TD3 showcased fewer troughs but more significant oscillations. These findings contribute to a better understanding of the strengths and weaknesses of each algorithm in the context of the tested environment.

V. Discussion

As previously indicated in the introduction section, the original research by Krylov et al. (2020) associated with the gym environment employed in this study utilized the PPO algorithm. However, the outcomes of this study demonstrate that, under specific hyperparameters, TRPO exhibited greater efficacy in producing favorable results.

While TRPO displayed the highest overall compatibility with the environment, it is worth noting that several alternative algorithms also demonstrated strong performances. For instance, TD3 outperformed TRPO in terms of peak AER and exhibited commendable consistency post-convergence, albeit with more pronounced oscillations than TRPO.

A smaller value for the clipping range in TRPO resulted in more conservative policy updates, which made the algorithm more stable but potentially slowed down learning. A lower learning rate made learning more gradual and less prone to overshooting optimal policies but required more iterations for convergence. A less than one gamma value made the algorithm prioritize immediate rewards over long-term ones. A small entropy coefficient (ent_coef) encouraged more deterministic policies, which may have limited exploration but also led to more stable policies. Lastly, a greater value function (vf_coef) led to more conservative policy updates and a stronger focus on exploiting the current knowledge.

Nevertheless, it is essential to acknowledge that further investigations have the potential to alter these conclusions. A promising avenue for enhancing this study in the future involves expanding the range of values explored for each hyperparameter. In this study, we deliberately constrained each hyperparameter within recommended ranges, primarily due to the time-intensive nature of generating meaningful results with significantly larger ranges. In future research, utilizing more powerful computational resources or extending the time frame would facilitate the exploration of broader hyperparameter ranges, thereby encompassing a more expansive experimental space.

Another avenue for improving this study could be increasing the number of experimental trials. Conducting ten trials per algorithm does not wholly represent all potential hyperparameter combinations; a more extensive trial setup could yield a more optimal hyperparameter configuration. Again, with access to enhanced computational capabilities or an extended temporal scope, a broader array of hyperparameter combinations could be explored to identify the most effective configuration.

A third avenue for enhancing this study involves expanding the scope of experimentation to encompass a more comprehensive array of reinforcement learning algorithms. The current gym environment for Deep Brain Stimulation only functions with algorithms that work in continuous action spaces. In a future study, we could eliminate this limitation, enabling the exploration of a broader range of reinforcement learning algorithms; this could lead to the discovery of a more effective reinforcement learning algorithm for Deep Brain Stimulation.

VI. Conclusion

In this research study, we assessed the performance of six reinforcement learning algorithms within a simulated gym environment that replicates the Deep Brain Stimulation procedure for Parkinson's disease patients. The objective was to determine the algorithm with the greatest aptitude for Deep Brain Stimulation. The original gym-related study employed the PPO algorithm to showcase the environment's capabilities. Our study found that among the six algorithms subjected to ablation studies to fine-tune hyperparameters, TRPO had the greatest capability in simulating Deep Brain Stimulation. However, it is noteworthy that other algorithms demonstrated comparable results.

From a scientific perspective, this research contributes to the field of neurocomputational modeling and machine learning applications in neuroscience. By using reinforcement learning to optimize stimulation parameters, and then providing an empirical comparison for different algorithms in deep brain stimulation, we can more deeply understand the prevention of tremors and its response to algorithmic interventions.

On the technological front, this study shows how reinforcement learning can be applied to medical technology, particularly in neurostimulation devices. By considering the model and hyper parameters used in this study, future deep brain stimulation devices may improve their efficacy in mitigating Parkinson’s symptoms while reducing unintended side effects. Additionally, advancements in ablation studies—such as Optuna, which was used in this study—can further refine optimization for the usage of reinforcement learning algorithms in medical applications.

From a societal perspective, this research underscores the potential for artificial intelligence to enhance medical treatments and improve patients' quality of life. Parkinson’s disease affects millions worldwide, and while DBS is an established treatment, optimizing its efficacy through machine learning could reduce the trial-and-error process currently required to determine stimulation parameters. By making DBS more efficient and precise, this research has implications for reducing healthcare costs, minimizing surgical interventions, and expanding access to high-quality treatment for neurodegenerative diseases.

In the future, with more advanced computational resources or an extended timeframe, we aspire to broaden the scope of ablation studies and incorporate a more comprehensive array of algorithms. We hope this study can provide greater insight into the efficacy of various reinforcement learning algorithms on Deep Brain Stimulation and that the results of this study can be applied to impact patients with Parkinson’s disease positively.

References

[1] Beitz, J. M. (2014). Parkinson’s disease: a review. Front Biosci (Schol Ed), 6(1), 65-74.

[2] Meara, J., & Hobson, P. (2018). Epidemiology of Parkinson’s disease. Parkinson's Disease in the Older Patient, 30-38.

[3] Bergman, H., Raz, A., Feingold, A., Nini, A., Nelken, I., Hilla, B. P., ... & Reches, A. (1998). Physiology of MPTP tremor. Movement disorders, 13(S3), 29-34.

[4] Benabid, A. L., Pollak, P., Hoffmann, D., Gervason, C., Hommel, M., Perret, J. E., ... & Gao, D. M. (1991). Long-term suppression of tremor by chronic stimulation of the ventral intermediate thalamic nucleus. The Lancet, 337(8738), 403-406.

[5] McIntyre, C. C., Savasta, M., Walter, B. L., & Vitek, J. L. (2004). How does deep brain stimulation work? Present understanding and future questions. Journal of clinical neurophysiology, 21(1), 40-50.

[6] Popovych, O. V., & Tass, P. A. (2019). Adaptive delivery of continuous and delayed feedback deep brain stimulation-a computational study. Scientific reports, 9(1), 10585.

[7] Bonaccorso, G. (2017). Machine learning algorithms. Packt Publishing Ltd.

[8] Krylov, D., Tachet, R., Laroche, R., Rosenblum, M., & Dylov, D. V. (2020). Reinforcement learning framework for deep brain stimulation study. arXiv preprint arXiv:2002.10948.

[10] Q. Gao et al., "Model-Based Design of Closed Loop Deep Brain Stimulation Controller using Reinforcement Learning," 2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS), Sydney, NSW, Australia, 2020, pp. 108-118, doi: 10.1109/ICCPS48487.2020.00018.

[11] Krylov, D., Dylov, D. V., & Rosenblum, M. (2020). Reinforcement learning for suppression of collective activity in oscillatory ensembles. Chaos: An Interdisciplinary Journal of Nonlinear Science, 30(3).

[13] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.

[14] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., ... & Wierstra, D. (2015). Continuous control with deep

reinforcement learning. arXiv preprint arXiv:1509.02971.

[15] Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015, June). Trust region policy optimization. In International conference on machine learning (pp. 1889-1897). PMLR.

[16] Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018, July). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In International conference on machine learning (pp. 1861-1870). PMLR.

[17] Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., ... & Kavukcuoglu, K. (2016, June). Asynchronous methods for deep reinforcement learning. In International conference on machine learning (pp. 1928-1937). PMLR.

[18] Fujimoto, S., Hoof, H., & Meger, D. (2018, July). Addressing function approximation error in actor-critic methods. In International conference on machine learning (pp. 1587-1596). PMLR.

[19] Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019, July). Optuna: A next-generation hyperparameter optimization framework. In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining (pp. 2623-2631).