CSCI 566 - Final Project Report

Akhil Parvathaneni, Alexander Romanus, and Mehul Shetty
aaparvat@usc.edu, romanus@usc.edu, mehulshe@usc.edu
...

Abstract

Video games serve as complex and dynamic virtual environments that present significant challenges for training autonomous agents through reinforcement learning (RL). Among them, the classic platformer game Super Mario Bros. offers a high-dimensional state space, discrete action sets, and delayed rewards, making it an appealing testbed for evaluating the efficacy of various RL algorithms. In this study, we investigate the application of four RL methods—Deep Q-Learning (DQL), Double Deep Q-Learning (DDQL), Proximal Policy Optimization (PPO), and Intrinsic Curiosity—to train agents capable of playing Super Mario Bros. We leverage the OpenAI gym interface together with the gym-super-mario-bros environment to provide a stable and reproducible setting for agent training. While DQL and DDQL have been widely explored in the context of Atari benchmarks, we introduce PPO, a method proposed by OpenAI, and Intrinsic Curiosity to this domain as novel contributions. We compare three distinct hyperparameter configurations for each RL method, each favoring different levels of exploration or exploitation. Our work thus contributes to understanding the performance landscape of PPO and intrinsic curiosity in a challenging environment that to our knowledge has not been extensively studied with this method.

1 Introduction

Reinforcement learning has gained substantial traction as a framework for training agents to master tasks by interacting with complex, often high-dimensional, environments. The game Super Mario Bros. provides a particularly suitable environment for RL research: while having a straightforward, two-dimensional state space, different combinations of moves, such as double and wall jumping, provide a rich action space. Furthermore, the array of interactions Mario can have with other game characters, obstacles, objects, and power-ups gives a vast array of possible experiences during each of Mario's lives.

The goal of each level is straightforward—navigate Mario rightward through the level to reach the flagpole, which marks completion. Along this journey, Mario encounters various challenges and opportunities: hostile enemies like Koopa Troopas that can eliminate him on contact, environmental obstacles such as pipes that require precise jumping, and more. Upon reaching the flagpole, the level is complete and a new challenge begins.

Mario can earn game points by collecting coins, killing enemies, and most prominently, by completing levels. To provide a suitable framework for RL, we define a different scoring mechanism as defined by the package gym-super-mario-bros. We'll explain more in Sections 3 and 4.

Unlike simpler Atari games, where strategies may be learned through relatively straightforward policies, succeeding in Super Mario Bros. requires the agent to understand both immediate rewards, like collecting coins, and delayed rewards, like reaching the flagpole. In general, reinforcement learning methods are also quite sensitive to hyperparameters: many attempts are usually needed to

Preprint. Under review.

effectively balance exploration and exploitation, and to make sure this balance is maintained across learning episodes.

Mastering Super Mario Bros. serves as a benchmark for advancing RL algorithms. Solutions developed in this setting can potentially transfer to other domains that require navigating complex state and action spaces, from more sophisticated, 3-dimensional video games like Minecraft, to real-world robotics tasks. By investigating the performance of multiple RL algorithms, we gain insights into the algorithms' capabilities, limitations, and potential improvements. Understanding how DQL, DDQL, PPO, and Intrinsic Curiosity (IC) handle challenges such as delayed rewards, non-trivial action sequences, and complex observational inputs contributes to the broader effort of refining RL methods for diverse and realistic problems.

Early in our research plan, we considered asynchronous advantage actor-critic (A3C) methods. However, we have since opted to use PPO, a more recent policy gradient method known for its stable and efficient training characteristics, as introduced by Schulman et al. at OpenAI. PPO's relatively simple implementation and robustness have made it popular in many RL tasks, but its performance on Super Mario Bros. remains underexplored. We have also chosen to explore Intrinsic Curiosity, specifically the forward dynamics model as an intrinsic reward, due to its success in a variety of RL applications but lack of implementation with Super Mario Bros. Our goal, therefore, is not only to compare DQL and DDQL—value-based approaches that have shown promise in many gaming benchmarks—but also to investigate PPO and IC as novel contributions in the context of this game.

2 Related Work

Deep RL research has its roots in work demonstrating that convolutional neural networks combined with Q-learning can learn to play Atari games directly from pixels. Mnih et al.'s seminal paper, "Playing Atari with Deep Reinforcement Learning," introduced DQN, showing how a deep network trained to estimate Q-values from raw game screens could surpass human-level performance on several Atari titles. While this approach established a strong baseline and demonstrated the feasibility of pixel-based RL, it also highlighted overestimation bias in Q-learning, limited exploration strategies, and the need for stable training methods.

To address the overestimation bias, Double Deep Q-Learning (DDQL), as presented by Hasselt et al., decouples action selection from action evaluation by maintaining an online and a target network. This modification helps stabilize learning and leads to more accurate value estimates, improving performance across a variety of environments. Although DDQL reduces overestimation and provides more stable learning curves than DQL, it still relies primarily on epsilon-greedy exploration and requires careful hyperparameter tuning.

In parallel, policy gradient methods have evolved to handle continuous and high-dimensional action spaces effectively. One notable advancement is Proximal Policy Optimization (PPO) by Schulman et al. at OpenAI, which strikes a balance between performance and simplicity. PPO simplifies trust region policy optimization methods by using a clipping mechanism to prevent large updates to the policy, improving training stability and sample efficiency. Despite widespread success in continuous control and complex tasks (e.g., robotics simulations and advanced gaming scenarios), the literature lacks extensive exploration of PPO in the Super Mario Bros. environment. By incorporating PPO into our research, we aim to extend the understanding of policy gradient methods beyond the Atari benchmarks and actor-critic frameworks previously investigated.

Furthermore, prior works have examined asynchronous training and multi-threaded approaches like A3C for faster exploration and improved state-space coverage. While we initially planned to incorporate such asynchronous methods, we ultimately focus on PPO to maintain methodological clarity and to contribute a novel perspective. The evaluation benchmarks employed in these seminal papers—often the 57 Atari games—are well-established. We follow a similar approach, using level completion and score in Super Mario Bros. As metrics to evaluate agent performance, while acknowledging that human-normalized scores and comparisons to publicly available repositories form critical aspects of validating our approach.

Another promising avenue in reinforcement learning for video games like Super Mario Bros. involves intrinsic curiosity modules (ICM) that leverage a forward dynamics model to enhance exploration. Pathak et al.'s ICM framework introduces an intrinsic reward signal based on the prediction error of a forward dynamics model, which attempts to predict the next state given the current state and action. This approach incentivizes the agent to explore novel and unpredictable areas of the state space, often leading to improved performance in sparse-reward environments like Super Mario Bros. However, while the intrinsic curiosity mechanism can effectively overcome exploration bottlenecks, it also introduces additional computational overhead due to the need for training auxiliary networks. Furthermore, the quality of the exploration largely depends on the capacity and accuracy of the forward dynamics model, which can struggle in high-dimensional and noisy environments, potentially leading to suboptimal exploration patterns. Despite these challenges, intrinsic curiosity remains a valuable tool for enhancing RL agents, especially in complex games requiring extensive exploration. In their paper, Pathak et al. used ICM along with A3C as its extrinsic reward, but we choose DDQN due to its stability in learning and less implementation complexity.

3 Methodology

Our methodology involves initially training agents using three distinct RL algorithms—DQL, DDQL, PPO—under three separate hyperparameter configurations each. This design choice allows us to systematically analyze how different settings of exploration and exploitation parameters affect performance. For each method, we implement a convolutional neural network (CNN) to evaluate each frame of Mario. In our RL experiments utilizing DQL and DDQL we then estimate each agent's action value function, Q(s, a), using the CNN. In using DQN, we both chose the action that maximized Q(s, a) and evaluated Q(s, a) based on that action. For the Double Deep Q Learning approach we mitigated a known issue with DQN: overestimation of the Q-value. This overestimation can lead to instability and divergence. Double DQN addresses this by decoupling the selection of the action with the evaluation of Q(s, a) based on the new action by delineating the process with two action-value functions, an online and a target version. An action is selected in the new state by doing an argmax of Q_{online}(s, a) over all possible actions. This action is then passed to Q_{target} (s, a_{max}), which performs the evaluation.

For PPO (Proximal Policy Optimization), we implement an actor-critic model that uses separate networks for policy (actor) and value estimation (critic). The policy network outputs a probability distribution over possible actions for a given state, while the critic predicts the state value to guide updates. We optimize the policy using a clipped surrogate objective to ensure stable updates by constraining changes to the policy within a predefined trust region. Generalized Advantage Estimation (GAE) is utilized to reduce variance in advantage calculations while maintaining low bias. Hyperparameters such as the clip range (ϵ), learning rate, discount factor (γ), and GAE lambda (τ) are varied across experiments to understand their impact on agent performance. Additionally, our PPO employs minibatch stochastic gradient descent over multiple epochs for each training step, enabling efficient use of experience samples. This approach ensures a balance between exploration and exploitation, adapting dynamically based on environmental feedback.

Finally, we will implement a method that integrates intrisic curiosity-driven exploration into a value-based RL pipeline. While standard Deep Q-Network (DQN) methods rely on extrinsic rewards derived directly from the environment (e.g., score increments, level completion), our approach augments the standard extrinsic reward with an intrinsic reward component. This intrinsic reward is generated from a learned forward dynamics model that predicts the agent's next state given its current state and selected action. By measuring the prediction error of the forward model, we produce a curiosity-driven incentive that encourages the agent to explore novel states, even in the absence of immediate environmental rewards. The network architecture will consist of the usual Q-networks used in DDQN and a Forward Dynamics model.

The forward dynamics model takes as input the current state and a one-hot encoded action. It encodes the input through a series of convolutional layers to extract latent features, combines these features with the action, and then decodes the resulting representation back into a predicted next-state image. A mean-squared error loss between the predicted and actual next state is used to train this model, thereby encouraging the forward model to learn accurate state transitions. Intuitively, states that the forward model cannot predict accurately are considered "novel" or "surprising," granting the agent an intrinsic incentive to explore them. This intrinsic reward is scaled by a hyperparameter and added to the extrinsic environmental reward before updating the Q-network.

At time step t, if s_t is the current state, a_t is the action taken, and s_{t+1} is the next state, $Q(s, a; \theta)$ is the online Q-network, $Q_{target}(s, a; \theta^-)$ is the target Q-network, and $F(s, a; \phi)$ is the forward dynamics model, γ is the discount factor, β is the intrinsic reward coefficient, and r_t and r_t^{int} are the extrinsic and intrinsic rewards then the Q-learning loss will be:

$$\mathcal{L}_Q(\theta) = \mathbb{E}_{(s,a,r,s',d)\sim D}\left[(Q(s, a; \theta) - y)^2\right]$$

d indicates if the next state is terminal. The forward dynamics loss is calculated as:

$$\mathcal{L}_{\text{Forward}}(\phi) = \mathbb{E}_{(s,a,s')\sim D}\left[\|F(s, a; \phi) - s'\|^2\right]$$

So the total loss becomes:

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_Q(\theta) + \beta \cdot \mathcal{L}_{\text{Forward}}(\phi)$$

In summary, our approach provides a structured framework to investigate how modifications in hyperparameters influence the performance of DQL, DDQL, and PPO in the Super Mario Bros. domain. By carefully selecting three sets of hyperparameters—favoring exploration, a balanced approach, and exploitation—we aim to understand not only the strengths and weaknesses of each algorithm but also how best to tune them for the intricate dynamics of this challenging environment.

3.1 Challenges and Trade-offs Summary

Each reinforcement learning approach we investigated presents advantages and challenges in the Super Mario Bros. environment. DQL and DDQL offer relatively straightforward implementations and benefit from extensive validation across various gaming environments, making them reliable baseline approaches. However, they suffer from some limitations: DQL exhibits overestimation bias in Q-value computation, while both methods show high sensitivity to hyperparameter choices, particularly in exploration rates and learning schedules. PPO, while demonstrating more stable training characteristics and consistent policy improvements through its clipping mechanism, demands substantially higher computational resources and requires a more complex implementation due to its actor-critic architecture and policy gradient computations. Our fourth approach, incorporating intrinsic curiosity, shows particular promise in Super Mario Bros.' sparse reward structure by encouraging exploration of novel states through its curiosity signal. However, this comes at the cost of increased network complexity, requiring an additional forward dynamics model and introducing extra training overhead to maintain both the curiosity and policy networks.

4 Experiments

4.1 Experimental Setup

All experiments were conducted using the OpenAI Gym environment (gym-super-mario-bros v3.0.6) running on Super Mario Bros. The state space was preprocessed by converting game frames to 84x84 grayscale images, with frame stacking of 4 consecutive frames to capture temporal information. The action space was simplified to 7 discrete actions: ['NOOP', 'right', 'right A', 'right B', 'right A B', 'A', 'left']. The agent gets rewarded for moving closer to the flag, killing enemies, collecting coins, and staying alive in each frame.

4.2 Hyperparameter Configurations and Results

For DQL and DDQL, each hyperparameter configuration differs primarily in gamma (discount factor), learning rate, and epsilon decay schedule. The first configuration leans toward exploration by using a higher learning rate and a slower epsilon decay, ensuring the agent tries more actions in the early and mid-training stages. The second configuration represents a middle ground, balancing exploration and exploitation with moderate γ and decay rates. The third configuration emphasizes exploitation, employing a lower ϵ end value and a higher γ to focus more on long-term returns and stable policies once initial exploration has provided sufficient experience.

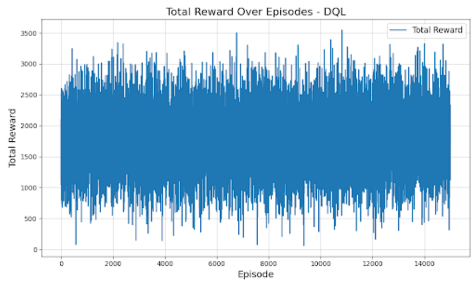
Similarly, for PPO, we employ three configurations with analogous intentions. The first configuration encourages broader exploration by using a higher learning rate and a larger clipping parameter, facilitating larger policy updates early in training. The second configuration uses a more moderate approach, with a reduced learning rate and slightly stricter policy update constraints to stabilize training. The third configuration, with a lower β (entropy coefficient) and smaller ϵ, fosters more conservative updates and thus more exploitation, as the agent refines its policy.

Table 1: Hyperparameter configurations for DQN/DDQL and PPO. The first configuration emphasizes exploration, the second configuration represents a middle ground, and the third configuration favors exploitation.

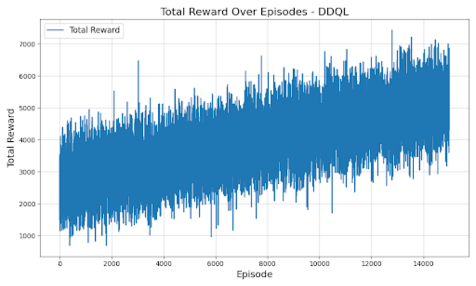
Method	**BATCH_SIZE**	**NUM_EPOCHS**	**GAMMA**	**LEARNING_RATE**	**EPS_START**	**EPS_END**	**EPS_DECAY**
				DQL/DDQL Configurations			
1st config	64	2	0.9	$1e^{-3}$	1.0	0.2	0.9995
2nd config	64	2	0.95	$1e^{-4}$	1.0	0.15	0.999
3rd config	64	2	0.99	$1e^{-4}$	1.0	0.05	0.99
				PPO Configurations			
Method	**BATCH_SIZE**	**NUM_EPOCHS**	**GAMMA**	**LEARNING_RATE**	**BETA**	**TAU**	**EPS_PPO**
1st config	64	2	0.9	$1e^{-3}$	0.1	0.95	0.5
2nd config	64	2	0.95	$1e^{-4}$	0.01	0.95	0.3
3rd config	64	2	0.99	$1e^{-4}$	0.001	0.95	0.1

For DQL/DDQL the third configuration seemed to work best which means by exploiting rewards slower it learns better. For PPO, the third configuration also worked best which has a low β and smaller ϵ. We then continued training for DQL, DDQL, and PPO using these hyperparameters for 15000 episodes. We then saved the total rewards and graphed it.

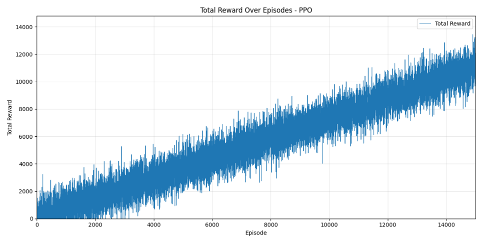
For our model that combines intrinsic curiosity with DDQN we lacked time and resources for hyperparameter tuning. So we opted to adopt a literature-informed default value for the intrinsic curiosity coefficient without disproportionately extending the experimental timeline. In Pathak et. al, the strength of the coefficient was 0.1. So we also chose this value for our experiment.



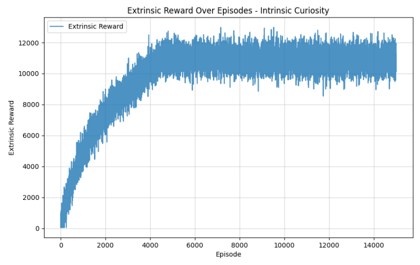
(a) DQL



(b) DDQL



(c) PPO



(d) DDQL + Intrinsic Curiosity

Figure 1: Comparison of Results Across Different Models

5 Discussion

The experimental results suggest several key insights into the relative performance and stability of the investigated reinforcement learning algorithms—DQL, DDQL, PPO, and the incorporation of intrinsic curiosity—when confronted with the complexities of Super Mario Bros. Observing the learning curves, one can see that both PPO and the Intrinsic Curiosity–enhanced DDQL consistently outpaced their more traditional Q-learning counterparts (DQL and DDQL without curiosity) in terms of achieved total rewards.

DQL did not seem to improve much over the course of its training time. The results were also extremely volatile. While DDQL did improve over time, its progress often appeared slightly volatile reflecting the well-documented fragility and hyperparameter sensitivity of value-based methods in large or complex environments. By contrast, PPO's policy gradient approach and the clipping mechanism appeared to grant a more stable ascent, indicative of the algorithm's built-in safeguards against overly large updates and its ability to capitalize on useful state-action sequences. The rewards also higher. This does not necessarily mean that agents went significantly further in PPO than DDQL but the rate of increase in rewards seems much sharper and much less noisy in PPO indicating better and faster learning. Similarly, the intrinsic curiosity module injected into DDQL evidently helped drive the agent toward novel and potentially rewarding states, mitigating the stall in exploration and enhancing the consistency and magnitude of the obtained returns.

These observations largely match our expectations. PPO's theoretical promise of more stable and efficient updates translated, in practice, into better results in a challenging environment like Super Mario Bros. Additionally, the improved exploration afforded by intrinsic curiosity aligns with prior literature, suggesting that curiosity-inspired intrinsic rewards can indeed bolster performance in environments where extrinsic rewards are sparse or difficult to obtain. While the successful demonstration of curiosity-driven exploration and PPO's stable performance are promising, it remains uncertain whether these conclusions hold across all platform-style tasks or generalize straightforwardly to other game genres and complex real-world scenarios. Differences in environmental dynamics, reward structures, and state representations can lead to variance in performance, so the positive trends we have observed should be considered domain-specific but indicative of each method's strengths. Still, the consistency of our results lends confidence that PPO and intrinsic curiosity can serve as valuable tools for scaling RL beyond more conventional or simpler Atari-style tasks.

6 Conclusion

Our experiments highlight both the promise and limitations that exist with state-of-the-art RL methods. They confirm that while simple value-based methods like DDQL can yield improvements over time, their susceptibility to hyperparameter choices and instability often impedes rapid or robust learning. Intrinsic curiosity, combined with DDQL, has one immediate and obvious advantage: it offers far superior exploration, much less likely to converge onto a prematurely suboptimal policy. PPO also somewhat demonstrated that policy gradient methods could, with careful clipping and advantage estimation, handle complexity such as Super Mario Bros. much more graciously than pure value-based methods, yielding more reliable and substantial performance gains.

That is not to say our method does not have any weaknesses. Intrinsic curiosity comes at the cost of additional computational overhead and model complexity, requiring forward models and specialized training routines that are more involved than standard Q-learning algorithms. The relative success of PPO also comes with its own tuning demands: finding appropriate hyperparameters can still pose a challenge, and its performance advantage may diminish in radically different domains. In this regard, future research will need to continue exploring the challenging task of finding an efficient balance between exploration and exploitation by experimenting with more advanced curiosity modules or integrating trust-region methods into already robust policy-based algorithms. Nonetheless, our findings paint a bright picture: using state-of-the-art policy gradient techniques combined with intelligent exploration mechanisms, one can steer RL agents to excel in tasks that more closely reflect the complexity and nuance of real-world challenges. This outcome ultimately strengthens the case for RL as a powerful, adaptable methodology for a broad spectrum of applications.

References

[1] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). *Human-level control through deep reinforcement learning*. Nature, 518(7540), 529–533.

[2] Hasselt, H. V., Guez, A., & Silver, D. (2016). *Deep reinforcement learning with double Q-learning*. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (pp. 2094–2100). AAAI Press.

[3] Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., ... & Kavukcuoglu, K. (2016). *Asynchronous methods for deep reinforcement learning*. In Proceedings of the 33rd International Conference on Machine Learning (ICML) (pp. 1928–1937). PMLR.

[4] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 2017. [Online]. Available: https://arxiv.org/pdf/1705.05363