A Study of Reinforcement Learning Algorithms on the CartPole-v1 Environment

Hou Zheyan
Department of Statistics and Data Science
Southern University of Science and Technology
Shenzhen, China
12313551@mail.sustech.edu.cn

Zhao Xuanzhi
Department of Statistics and Data Science
Southern University of Science and Technology
Shenzhen, China
12311856@mail.sustech.edu.cn

Xing Yue
Department of Life Science
Southern University of Science and Technology
Shenzhen, China
12312632@mail.sustech.edu.cn

Abstract

This paper presents a comprehensive comparative study of three reinforcement learning algorithms on the classic CartPole-v1 control task. We implement and analyze REINFORCE, Advantage Actor-Critic (A2C), and Noisy Dueling Double DQN with Conservative Q-Learning extensions. Our experiments focus on algorithm stability, convergence properties, and the effects of key hyperparameters including learning rate, discount factor, and exploration mechanisms. We demonstrate that all implemented algorithms can achieve perfect performance (500±0 score) on CartPole-v1 with appropriate parameter tuning. The study includes detailed analysis of offline reinforcement learning performance using different dataset compositions (expert, mixed, and random) and provides insights into the trade-offs between exploration efficiency and learning stability in policy-based versus value-based methods.

1 Introduction

Reinforcement learning (RL) has demonstrated remarkable success in solving complex decision-making problems across various domains. The CartPole-v1 environment from the Gymnasium toolkit [Towers et al., 2023] serves as an ideal testbed for RL algorithm analysis due to its simplicity, well-defined dynamics, and clear success criteria. In this STA303 Artificial Intelligence course project, we implement and compare multiple RL algorithms to understand their fundamental characteristics and practical performance.

The CartPole-v1 task involves balancing a pole attached to a moving cart through a frictionless hinge. The agent must apply forces to the cart to prevent the pole from falling. The environment provides a 4-dimensional state space (cart position, cart velocity, pole angle, pole angular velocity) and a discrete action space (left or right force). Each time step the pole remains upright yields a +1 reward,

with episodes terminating when the pole deviates more than 12° from vertical, the cart moves beyond 2.4 units from center, or 500 steps are reached.

Our work explores three distinct algorithmic approaches: policy-based methods through REINFORCE, actor-critic methods via A2C, and value-based methods using an enhanced DQN variant. Beyond online learning, we investigate offline reinforcement learning with Conservative Q-Learning to understand how pre-collected data affects learning performance. The study emphasizes algorithmic trade-offs, hyperparameter sensitivity, and practical implementation considerations rather than merely reporting performance metrics.

2 Background and Related Work

2.1 Reinforcement Learning Fundamentals

The standard RL framework involves an agent interacting with an environment over discrete time steps. At each step t, the agent observes state $s_t \in \mathcal{S}$, takes action $a_t \in \mathcal{A}$, receives reward $r_t \in \mathbb{R}$, and transitions to next state $s_{t+1} \sim P(\cdot|s_t, a_t)$. The agent's goal is to learn a policy $\pi(a|s)$ that maximizes the expected discounted return $J(\pi) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_t]$.

2.2 Algorithm Categories

Our study spans three major RL algorithm categories, each representing a distinct philosophical approach to the reinforcement learning problem.

Policy gradient methods, exemplified by REINFORCE [Williams, 1992], directly optimize the policy using Monte Carlo estimates of returns. The policy gradient theorem provides the theoretical foundation: $\nabla_\theta J(\theta) = \mathbb{E}[\nabla_\theta \log \pi_\theta(a|s) G_t]$, where $G_t = \sum_{k=t}^{T} \gamma^{k-t} r_k$ is the return from time t. This approach is elegant in its simplicity but suffers from high variance in gradient estimates.

Actor-critic methods, such as A2C [Mnih et al., 2016], combine policy gradient with value function approximation to reduce variance. The actor updates the policy using $\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi_\theta(a_t|s_t) A(s_t, a_t)$, while the critic learns the value function $V(s)$ or advantage function $A(s, a)$. This dual architecture allows for more stable learning but introduces additional complexity.

Value-based methods, beginning with DQN [Mnih et al., 2015], learn a Q-function $Q(s, a)$ through temporal difference learning. We implement several enhancements including Double DQN [Van Hasselt et al., 2016] to reduce overestimation bias, Dueling networks [Wang et al., 2016] to separate state value and action advantage, and Noisy networks [Fortunato et al., 2018] for more intelligent exploration. These improvements address known limitations of vanilla DQN while maintaining its sample efficiency.

For offline reinforcement learning, we employ Conservative Q-Learning (CQL) [Kumar et al., 2020], which addresses distribution shift by adding a conservative regularization term: $\min_Q \alpha \mathbb{E}_{s\sim D}[\log \sum_a \exp(Q(s, a)) - \mathbb{E}_{a\sim D(a|s)}[Q(s, a)]] + \frac{1}{2}\mathbb{E}_{(s,a,s')\sim D}[(Q - \mathcal{B}^* Q)^2]$. This approach is particularly relevant for real-world applications where online interaction is costly or dangerous.

3 Methodology

3.1 Algorithm Implementations

Our REINFORCE implementation employs a policy network with stochastic action selection using categorical distribution. We calculate Monte Carlo returns with discount factor γ and apply return normalization ($\frac{G_t - \mu}{\sigma}$) to reduce variance. The algorithm uses episode-based training with batch size equal to one complete episode, reflecting its Monte Carlo nature.

For A2C, we implement a shared neural network feature extractor with separate policy (actor) and value (critic) heads. The algorithm operates in an on-policy manner, collecting complete episodes of experience, from which we compute advantages using Generalized Advantage Estimation (GAE) with parameter λ to trade off bias and variance in advantage estimation. An entropy regularization term $\beta H(\pi(\cdot|s))$ is incorporated into the loss function to promote exploration and prevent premature

convergence. Additionally, advantage normalization—standardizing advantages to zero mean and unit variance—is applied to reduce gradient variance and enhance training stability and consistency.

Our enhanced DQN variant, termed NoisyD3QN, combines multiple improvements: a dueling architecture that separates state value and action advantage as $Q(s,a) = V(s) + (A(s,a) - \frac{1}{|\mathcal{A}|}\sum_{a'} A(s,a'))$; double Q-learning to mitigate overestimation bias; and noisy networks that replace linear layers with NoisyLinear layers containing parameterized noise for automatic exploration. This integrated approach addresses several known limitations of traditional DQN.

3.2 Experimental Setup

All algorithms were trained on CartPole-v1 with base hyperparameters: maximum episode length of 500 steps, and target success criterion of average score ≥ 475 over 100 episodes.

For the online A2C agent, we adopt an episode-based training protocol: policy and value networks are updated once per complete episode. Based on empirical tuning, we set the GAE parameter $\lambda = 0.8$, learning rate 0.002, entropy coefficient 0.001, discount factor $\gamma = 0.99$, and gradient clipping with max norm 0.5. The shared network architecture consists of two hidden layers (64 units each) with ReLU activations, followed by separate policy and value heads.

For CQL experiments, we created three dataset types to study the effect of data quality: expert datasets containing 50,000 transitions from trained NoisyD3QN agents; mixed datasets with 50% expert and 50% random actions (45,198 transitions); and random datasets with 100% random actions (22,599 transitions). This stratification allows us to analyze how data composition affects offline learning performance.

We evaluate algorithms using multiple metrics: average score over 100 test episodes, success rate (percentage of episodes achieving 500 score), training stability measured by variance in learning curves, and convergence speed quantified as episodes to reach target performance.

4 Experimental Results and Analysis

4.1 REINFORCE Performance and Insights

REINFORCE demonstrates remarkable stability across hyperparameter configurations, with all tested settings eventually achieving perfect performance (500±0 score).

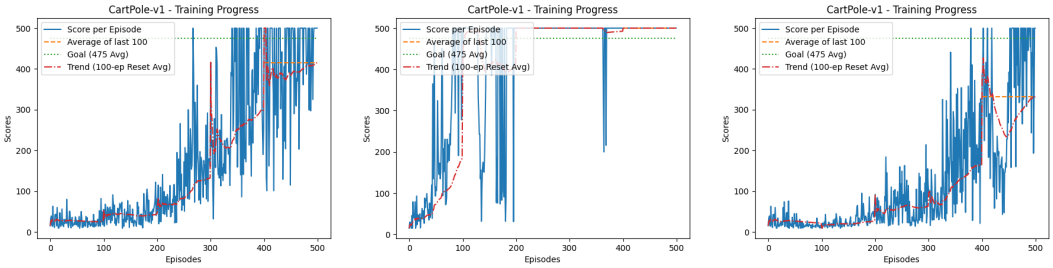
A higher learning rate (5e-3) accelerates the convergence process. Increasing the discount factor γ from 0.99 to 0.999 caused the REINFORCE algorithm to exhibit significantly higher variance and instability. This is because REINFORCE, as a Monte Carlo method, relies on the full-trajectory return G_t, whose variance is greatly amplified by a high γ, thereby destabilizing the learning process.

Table 1 contrasts REINFORCE with DQN from the perspective of algorithmic principles:

Table 1: REINFORCE vs. DQN characteristics

Dimension	**DQN**	**REINFORCE**
Learning Target	$Q(s,a)$ value function	$\pi(a\|s)$ policy function
Update Basis	TD target	Monte Carlo return
Data Usage	Experience replay	Episode buffer
Exploration	ϵ-greedy	Policy sampling
Stability Mechanism	Target network	Return normalization
Update Method	MSE minimization	Gradient ascent

Compared to DQN (Figure 3c), REINFORCE (Figure 2) exhibits fundamentally different characteristics. While DQN learns more quickly through experience replay and temporal difference learning, REINFORCE's episode-based approach provides excellent final performance with minimal hyperparameter tuning. This contrast highlights the tension between sample efficiency and algorithmic stability. REINFORCE's inherent stochasticity from policy sampling provides natural exploration, eliminating the need for explicit exploration mechanisms like ϵ-greedy.



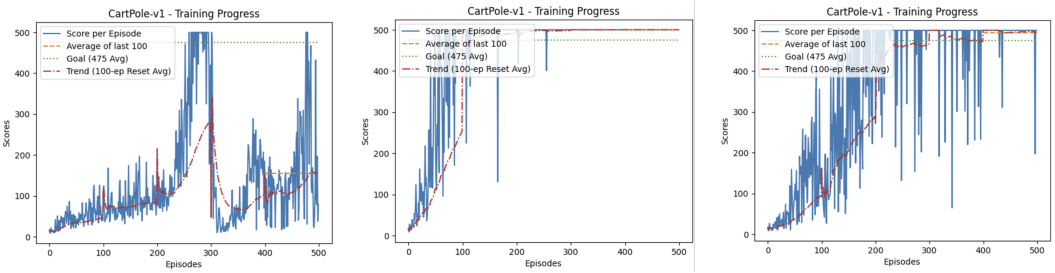
(a) Default (b) Higher LR (lr = 5e-3) (c) Higher γ ($\gamma = 0.999$)

Figure 1: REINFORCE training performance with different hyperparameters.

4.2 A2C Performance and Parameter Analysis

A2C's performance on `CartPole-v1` is highly sensitive to the design of its advantage estimation and regularization mechanisms. Central to this sensitivity is the Generalized Advantage Estimation (GAE) parameter λ, which controls the bias-variance trade-off in advantage computation. While higher values (e.g., $\lambda = 0.95$) reduce bias by incorporating longer-horizon returns, they also amplify variance—particularly problematic when the value function is still poorly calibrated early in training. Conversely, very low values (e.g., $\lambda = 0.1$) produce low-variance but high-bias estimates that over-rely on an initially inaccurate critic, leading to unstable learning dynamics and large fluctuations in episode returns.

Through systematic empirical evaluation, we found that $\lambda \in [0.4, 0.8]$ yields the most reliable convergence on `CartPole-v1`, with $\lambda = 0.8$ consistently achieving the fastest learning without oscillation. This range reflects a sweet spot where the algorithm benefits from moderate credit assignment depth while maintaining numerical stability—a balance particularly important in simple, deterministic environments where excessive lookahead provides little additional signal.



(a) GAE-$\lambda = 0.1$ (b) GAE-$\lambda = 0.8$ (c) GAE-$\lambda = 0.95$

Figure 2: Training curves for different λ values.

Entropy regularization plays a secondary but non-negligible role. Contrary to common practice in more complex domains, `CartPole`'s dense reward structure requires only minimal exploration encouragement. We observed that entropy coefficients in the range $[0.001, 0.01]$ maintain sufficient stochasticity to avoid premature convergence while preserving rapid policy improvement. In contrast, larger values such as 0.1—often used in sparse-reward settings—introduce unnecessary randomness that degrades performance and delays convergence to the optimal 500-step horizon.

Critically, two implementation-level enhancements proved indispensable: advantage normalization and gradient clipping. Normalizing advantages to zero mean and unit variance dramatically stabilizes policy gradient updates, effectively decoupling learning dynamics from the scale of rewards. Combined with gradient clipping at a max norm of 0.5, these techniques prevent the catastrophic divergence occasionally observed in naive A2C implementations. Together, they transform A2C from a theoretically elegant but fragile method into a robust and efficient online learner for low-dimensional control tasks like `CartPole`.

These findings underscore a broader principle: even "standard" algorithms like A2C require careful attention to both hyperparameter selection and numerical implementation details to achieve reliable performance. The interplay between GAE tuning, entropy scheduling, and optimization safeguards reveals that practical success often hinges more on engineering rigor than on algorithmic novelty alone.

4.3 NoisyD3QN: Enhanced Exploration and Architecture

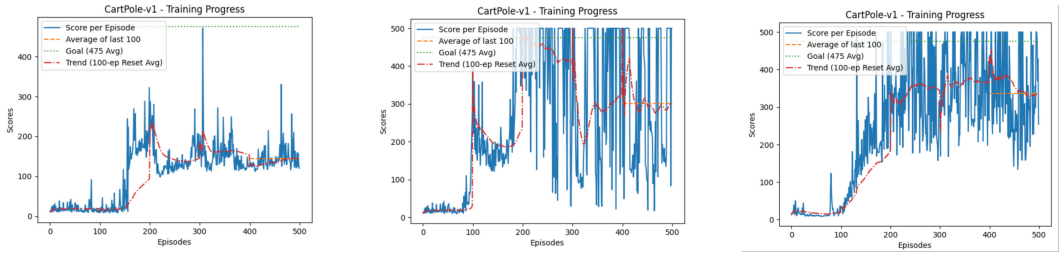
NoisyD3QN represents a significant enhancement over the standard DQN framework, incorporating three key innovations that collectively address fundamental limitations of value-based reinforcement learning. First, it employs a **Dueling network architecture** that separates state value estimation from action advantage calculation, expressed as $Q(s,a) = V(s) + (A(s,a) - \frac{1}{|\mathcal{A}|}\sum_{a'} A(s,a'))$. This architectural modification provides more stable learning by independently modeling what states are valuable versus which actions are advantageous within those states.

Second, the algorithm implements **Double DQN** target calculation to mitigate the overestimation bias inherent in standard Q-learning. Rather than using the target network for both action selection and evaluation, Double DQN employs the online network to select actions and the target network to evaluate them: $y = r + \gamma Q_{\text{target}}(s', \arg\max_{a'} Q(s', a'))$. This decoupling significantly improves learning stability.

Third, and most innovatively, NoisyD3QN replaces traditional exploration mechanisms with **Noisy Networks**. Instead of using ϵ-greedy exploration with manually tuned decay schedules, noisy networks inject parameterized noise directly into the network weights. This enables automatic, adaptive exploration that evolves naturally during training as the noise parameters are learned alongside the Q-values.

Our implementation required several code modifications: replacing standard linear layers with NoisyLinear layers throughout the network architecture, removing ϵ-greedy scheduling entirely, and modifying the dueling network to accommodate noisy parameters. These changes transformed exploration from an externally controlled parameter to an intrinsic property of the learning system.

During initial experiments with parameters matching standard DQN (500 training episodes, fixed random seed), NoisyD3QN exhibited surprisingly poor training performance, rarely achieving high scores. We discovered that the algorithm's conservative value estimation and additional stochasticity from noisy networks required adjusted hyperparameters. By reducing the experience replay buffer size from 50,000 to 10,000 transitions, we accelerated learning through faster experience turnover. Accelerating target network updates from every 500 to every 100 steps provided more timely feedback. Increasing batch size from 32 to 64 offered stabilizing effects against the algorithm's inherent variance.



(a) D3QN-Initial parameters (b) D3QN-Better parameters (c) DQN-Better parameters

Figure 3: Parameter Analysis and Performance Comparison

Interestingly, these adjustments revealed a fundamental insight about NoisyD3QN's learning dynamics. While the algorithm displays greater training variance compared to DQN—appearing less stable during learning—this actually reflects its more sophisticated exploration strategy. The noisy networks continuously probe the action space even as the policy converges, preventing premature exploitation of suboptimal strategies. This explains why, despite seemingly erratic training curves, NoisyD3QN achieved perfect scores (500±0) in all 100 evaluation episodes, demonstrating exceptional generalization capabilities.

The apparent contradiction between training instability and testing excellence highlights a key advantage of noisy networks: they maintain exploration as an integral part of the learning process rather than treating it as an external parameter to be annealed. This results in policies that are not only high-performing but also robust, having been exposed to a wider variety of state-action pairs during training. Our findings suggest that traditional metrics of training stability may be misleading for algorithms with sophisticated exploration mechanisms, and that test performance ultimately validates the learning approach.

4.4 Offline Reinforcement Learning with Conservative Q-Learning

Offline reinforcement learning presents unique challenges distinct from online methods, primarily the issues of value overestimation and distributional shift between the behavior policy that collected the data and the learned policy. We address these challenges using Conservative Q-Learning (CQL), an algorithm specifically designed to learn conservative yet effective policies from pre-collected datasets without additional environmental interaction.

The core innovation of CQL lies in its conservative regularization term added to the standard Bellman error. The complete loss function is expressed as:

$$\min_Q \alpha \mathbb{E}_{s\sim D}\left[\log \sum_a \exp(Q(s,a)) - \mathbb{E}_{a\sim D(a|s)}[Q(s,a)]\right] + \frac{1}{2}\mathbb{E}_{(s,a,s')\sim D}\left[(Q - \mathcal{B}^*Q)^2\right]$$

where $\alpha > 0$ controls the trade-off between conservatism and Bellman error minimization. The first term constitutes the conservative penalty: $\log \sum_a \exp(Q(s,a))$ pushes up Q-values for all actions, while $\mathbb{E}_{a\sim D(a|s)}[Q(s,a)]$ pushes down Q-values specifically for actions observed in the dataset. This asymmetric regularization consciously underestimates Q-values for out-of-distribution actions while maintaining reasonable estimates for in-distribution actions, ultimately learning safer and more reliable policies.

To evaluate CQL's robustness to data quality, we constructed three distinct datasets with varying coverage and optimality:

- **Expert Dataset:** 50,000 transitions collected from a fully trained NoisyD3QN agent, representing high-quality but potentially narrow coverage.
- **Random Dataset:** 22,599 transitions generated entirely through random action selection, providing broad state coverage but poor reward signals.
- **Mixed Dataset:** 45,198 transitions combining 50% expert and 50% random actions, balancing quality with exploration diversity.

During initial training with default parameters, we encountered numerical instability as Q-values diverged to large negative magnitudes. We addressed this through two key implementation techniques. First, we employed the **log-sum-exp trick** for numerical stability: $\log \sum \exp(Q) = \max(Q) + \log \sum \exp(Q - \max(Q))$. Second, we progressively increased the conservative coefficient α with dataset quality: $\alpha = 6.0$ for expert data, $\alpha = 10.0$ for mixed data, and $\alpha = 14.0$ for random data, with corresponding training epochs of 80, 120, and 180 respectively.

Our experimental results reveal fascinating learning dynamics. For both expert and mixed datasets, the average loss exhibited a characteristic pattern: initial rapid decrease as the algorithm learned from the data, followed by a gradual increase as the conservative regularization became dominant. This pattern reflects CQL's two-phase learning process—first acquiring knowledge from the dataset, then consciously downweighting potentially unreliable Q-value estimates.

The algorithm demonstrated remarkable performance on expert data, achieving perfect scores (500±0) in all 100 test episodes. More interestingly, the resulting policies exhibited qualitatively different behavior than their online counterparts. Visual observation of the CartPole animation revealed smoother, more natural control strategies with reduced oscillation, suggesting that conservative regularization not only improves safety but also leads to more graceful policies.

Perhaps most impressive was CQL's performance on the mixed dataset. Despite containing 50% random actions with poor reward signals, the algorithm maintained near-perfect performance (496.2±38.1 average score, 99% success rate). This resilience highlights CQL's effectiveness as a pessimistic

offline method—by conservatively estimating Q-value lower bounds, it naturally filters out noise while retaining useful knowledge from the expert portion of the data.

Even with purely random data, CQL managed to learn functional policies (414.7±121.9 average score, 55% success rate), demonstrating its ability to extract meaningful patterns from suboptimal datasets. The performance gradient across datasets—expert > mixed > random—cleanly illustrates the fundamental trade-off in offline RL: performance is ultimately bounded by data quality, but sophisticated algorithms like CQL can approach this bound more closely.

These findings have important implications for practical applications of reinforcement learning. CQL's ability to learn effective policies from mixed-quality data suggests promising applications in real-world domains where perfectly optimal demonstrations are unavailable, but diverse experience exists. The algorithm's conservative nature makes it particularly suitable for safety-critical applications where overestimation could lead to catastrophic failures.

5 Discussion and Insights

5.1 Algorithmic Trade-offs and Design Philosophy

Our comparative study reveals fundamental trade-offs embedded in each algorithmic approach. REINFORCE's simplicity stems from its direct policy optimization, but this comes at the cost of sample inefficiency—each gradient update requires complete episode trajectories. Interestingly, this apparent weakness becomes a strength in environments where data collection is cheap but computational resources for hyperparameter tuning are limited. The algorithm's stability across diverse hyperparameter settings suggests that its high-variance gradient estimates are surprisingly robust in practice.

A2C's key advantage lies in its variance reduction through the critic network, which estimates state values to compute more stable advantage estimates than REINFORCE's full Monte Carlo returns.The entropy regularization coefficient, for instance, embodies a direct trade-off between exploration and exploitation. Our finding that low coefficients (0.001) work best for CartPole indicates that simple environments with dense rewards require minimal entropy regularization to maintain exploration without destabilizing learning

NoisyD3QN presents a fascinating case of architectural evolution addressing algorithmic limitations. The transition from manual ϵ-greedy exploration to learned noise parameters represents more than just implementation convenience—it fundamentally changes how exploration is conceptualized. Rather than treating exploration as an external process to be annealed, noisy networks integrate it into the learning mechanism itself. This explains the counterintuitive observation that higher training variance correlates with better test performance: the "instability" during training reflects continuous exploration, not poor learning.

CQL's performance gradient across datasets illuminates a fundamental truth about offline RL: the algorithm can only be as good as its data. However, the steepness of this gradient—from perfect performance on expert data to moderate performance on random data—suggests that sophisticated algorithms can extract substantial value even from imperfect data. The algorithm's conservative bias, while reducing performance on optimal data, provides crucial robustness when data quality is uncertain.

5.2 Practical Implementation Considerations

Beyond theoretical comparisons, our experiments highlight practical considerations often overlooked in algorithmic descriptions. REINFORCE's return normalization, while theoretically optional, proved essential for stable learning—a detail that significantly impacts real-world usability. Similarly, A2C's advantage normalization and gradient clipping, though not part of the core algorithm, were indispensable for preventing training divergence.

The parameter sensitivity patterns we observed have important implications for real-world deployment. REINFORCE's robustness to hyperparameter choices makes it ideal for applications where extensive tuning is impractical. Conversely, DQN variants' sensitivity suggests they may require automated hyperparameter optimization for reliable performance, adding computational overhead that must be factored into deployment decisions.

CQL's numerical stability issues with the log-sum-exp term serve as a cautionary tale about the gap between mathematical formulations and practical implementations. The need for stabilization techniques even in a simple environment like CartPole suggests that these challenges would be magnified in more complex domains.

5.3 Limitations and Future Directions

While CartPole-v1 provides an excellent pedagogical testbed, its simplicity necessarily limits the generalizability of our findings. The environment's low-dimensional state space, deterministic dynamics, and sparse reward structure represent best-case conditions that may not reflect challenges in more complex domains. Future work should validate these algorithms on environments with partial observability, stochastic transitions, and dense reward structures to better understand their scaling properties.

The perfect scores achieved by all algorithms ultimately reveal more about the environment than the algorithms themselves. In more challenging domains, we would expect clearer performance differentiation based on sample efficiency, exploration strategy, and robustness to hyperparameter choices. This suggests that benchmarking on "solved" environments like CartPole may obscure important algorithmic differences that become critical in unsolved problems.

An interesting direction for future research would be hybrid approaches that combine insights from multiple algorithmic families. For instance, incorporating noisy network exploration into policy gradient methods, or applying conservative regularization to online algorithms, might capture the strengths of each approach while mitigating their weaknesses.

6 Conclusion

This study provides a comprehensive analysis of reinforcement learning algorithms through the lens of CartPole-v1, revealing not only their performance characteristics but also the philosophical differences underlying their design. We demonstrate that all major algorithmic approaches—policy gradient, actor-critic, value-based, and offline methods—can achieve optimal performance on this classic benchmark given appropriate implementation and tuning.

Our key contribution lies in elucidating the practical trade-offs that govern algorithm selection for real-world applications. REINFORCE offers simplicity and stability at the cost of sample efficiency, making it suitable for data-rich but compute-limited scenarios. A2C provides a balanced middle ground, though with increased implementation complexity. NoisyD3QN represents the culmination of value-based innovations, achieving high performance through architectural sophistication. CQL enables effective learning from pre-collected data, with performance tightly coupled to data quality.

Beyond specific algorithmic comparisons, this work highlights several broader insights. First, implementation details often matter as much as algorithmic choices—normalization, clipping, and numerical stability techniques proved essential across all methods. Second, evaluation metrics must be carefully chosen; training stability and test performance can tell conflicting stories, as demonstrated by NoisyD3QN. Third, environment characteristics should guide algorithm selection more than absolute performance metrics.

Looking forward, the field would benefit from more nuanced evaluation frameworks that consider not only final performance but also sample efficiency, hyperparameter sensitivity, implementation complexity, and computational requirements. As reinforcement learning moves from research labs to real-world applications, these practical considerations will become increasingly important.

Ultimately, this study reinforces that there is no single "best" reinforcement learning algorithm—only algorithms best suited to particular constraints, environments, and objectives. The art of reinforcement learning practice lies in understanding these trade-offs and selecting the right tool for each unique challenge.

Acknowledgments

This research was conducted as part of the STA303 Artificial Intelligence course project (2025-2026). We thank the course instructors for their guidance and feedback.

References

Meire Fortunato, Mohammad Gheshlaghi Azar, Bilal Piot, Jacob Menick, Ian Osband, Alex Graves, Vlad Mnih, Remi Munos, Demis Hassabis, Olivier Pietquin, et al. Noisy networks for exploration. *International Conference on Learning Representations*, 2018.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. *International conference on machine learning*, pages 1928–1937, 2016.

Mark Towers, Jordan Terry, Ariel Kwiatkowski, John Balis, Gianluca Cola, Rodrigo de Lazcano, Kai Lee, Sami Ö. Üstün, and Markus Ritter. Gymnasium. `https://github.com/Farama-Foundation/Gymnasium`, 2023. Version 0.29.1.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. *Proceedings of the AAAI conference on artificial intelligence*, 30(1), 2016.

Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. Dueling network architectures for deep reinforcement learning. *International conference on machine learning*, pages 1995–2003, 2016.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.