FAIReport: Comparative Analysis of Tabular and Deep RL in Stochastic Grids

Rupak Banerjee (22110226), Brijesh Dabhi (23110067), Parth Shah (21110152)
Indian Institute of Technology (IIT) Gandhinagar
November 2025

***Abstract*—This paper presents a rigorous comparative analysis of Reinforcement Learning (RL) algorithms across varying degrees of environmental complexity. We evaluate tabular methods (Monte Carlo, SARSA, Q-Learning, Double Q-Learning) and Deep Reinforcement Learning methods (DQN, Double DQN) on stochastic grid-world environments. Our experiments reveal a significant performance divergence: while SARSA exhibits superior stability in small-scale environments (7×7), Monte Carlo (MC) demonstrates significantly better convergence in complex, sparse-reward settings (25×25) due to its ability to propagate rewards over long horizons. Furthermore, we analyze the impact of maximization bias in Q-Learning and the efficacy of Deep Q-Networks in handling large state spaces.**

***Index Terms*—Reinforcement Learning, SARSA, Monte Carlo, Q-Learning, Deep Q-Networks, Double Q-Learning.**

I. Introduction

Reinforcement Learning (RL) involves agents learning optimal policies through trial and error interaction with an environment. The trade-off between bias and variance, as well as the challenge of credit assignment in sparse reward environments, remains a core area of research.

This study investigates three distinct comparative phases:

1) **On-Policy Evaluation:** Monte Carlo vs. SARSA in both simple and complex stochastic grids.
2) **Off-Policy Bias Reduction:** Q-Learning vs. Double Q-Learning (DQL).
3) **Function Approximation:** Deep Q-Networks (DQN) vs. Double DQN (DDQN).

II. Methodology

A. Environments

Two environments were designed to test the agents' robustness:

1) Simple Grid (7×7): A discrete grid where the agent must reach a static center goal (+50 reward) while avoiding two moving enemies (-20 reward) and a teleporter. The state space is small ($7^2 \times$ enemy_pos $\approx$ tractable).

2) Complex Grid (25×25): A significantly larger state space featuring:

- **Layout:** Two concentric diamond wall rings with cardinal openings.
- **Adversaries:** Three enemies circling at Manhattan radii 3, 5, and 7.
- **Stochasticity:** Teleporters with probabilistic destinations (Goal/Enemy/Random).
- **Rewards:** Goal (+1000), Enemy (-200), Step (-1).

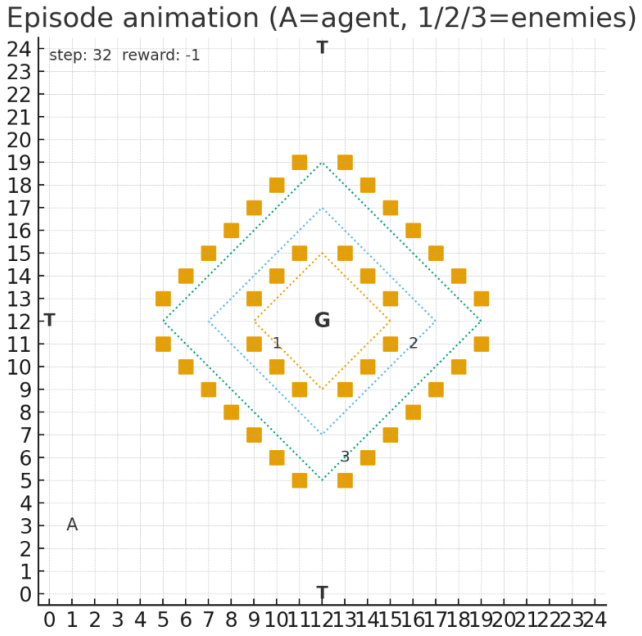

Fig. 1: Complex 25×25 grid environment showing the agent (A), enemies (1/2/3), walls, and teleporter locations.

B. Algorithms

1) Monte Carlo (MC): MC methods wait until the end of an episode to update value estimates. The update rule for value $V(S_t)$ is:

$$V(S_t) \leftarrow V(S_t) + \alpha[G_t - V(S_t)] \tag{1}$$

where G_t is the actual return following time t.

2) SARSA: An on-policy TD(0) method that updates values based on the next action actually taken:

$$Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma Q(S', A') - Q(S, A)] \tag{2}$$

3) Q-Learning & Double Q-Learning: Standard Q-learning uses the max operator, leading to maximization bias. Double Q-Learning decouples selection and evaluation using two estimators, Q_1 and Q_2.

4) Deep Q-Networks (DQN): DQN approximates the Q-function using a neural network, minimizing the loss:

$$L(\theta) = \mathbb{E}[(r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta))^2] \tag{3}$$

III. Experimental Results

A. Phase I: Monte Carlo vs. SARSA

1) Simple Environment Results: In the 7×7 grid, SARSA outperformed Monte Carlo slightly in terms of convergence speed and final success rate.

- **SARSA Success:** 93.3%
- **MC Success:** 89.5%

SARSA's step-by-step bootstrapping allowed it to learn from partial episodes, reducing variance in the highly volatile small grid where enemy collisions are frequent.

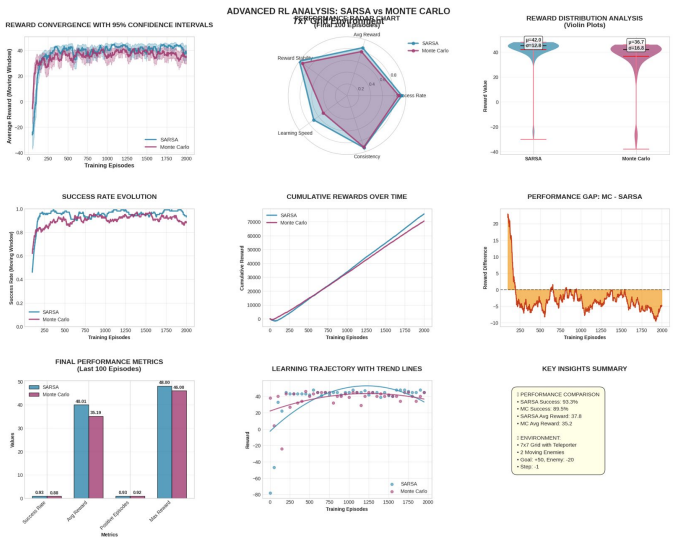

Fig. 2: **Simple 7x7 Grid:** SARSA (Blue) shows slightly faster convergence and higher stability compared to Monte Carlo (Purple).

2) Complex Environment Results: In the 25×25 environment, a dramatic reversal in performance was observed.

- **MC Success:** 65.8% (Avg Reward: 426.4)
- **SARSA Success:** 20.6% (Avg Reward: -215.8)

Analysis: SARSA failed to propagate the sparse goal reward (+1000) effectively across the large state space. The "bootstrap" nature of SARSA means it relies on the value of the next state (S'). If $Q(S')$ is uninformative (which is true for most of the large grid early on), learning stalls. Conversely, Monte Carlo, despite high variance, performs a complete trace. Once an episode hits the goal, the high return G_t is backpropagated to all visited states immediately.

B. Phase II: Q-Learning vs. Double Q-Learning

Both algorithms were tested on the Complex 25×25 grid.

- **Q-Learning:** Success 14.0%, Avg Reward -236.7
- **Double Q:** Success 15.9%, Avg Reward -215.1

Double Q-learning exhibited slightly better stability by reducing overestimation bias, though both methods struggled due to dimensionality.

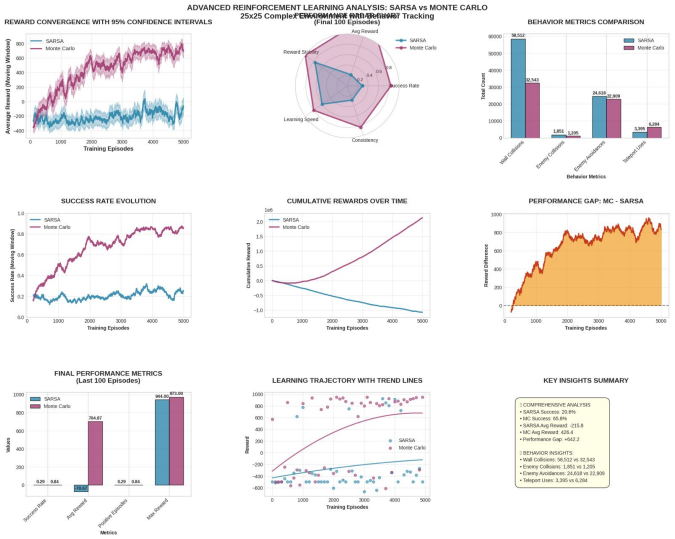

Fig. 3: **Complex 25x25 Grid:** Monte Carlo (Purple) significantly outperforms SARSA (Blue). SARSA struggles to overcome the negative step penalties and sparse rewards.

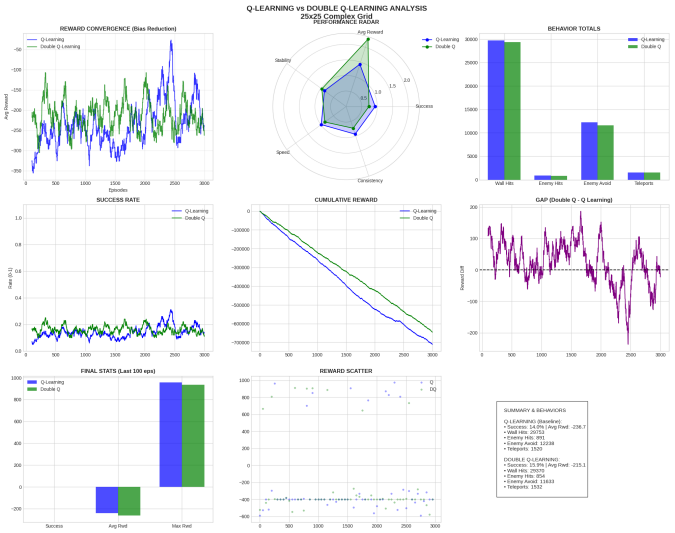

Fig. 4: **Q vs Double Q:** Both tabular methods struggled in the complex environment, but Double Q (Green) showed marginally better stability.

C. Phase III: Deep Q-Networks (DQN vs DDQN)

Using function approximation alleviated the dimensionality bottleneck.

- **DQN:** Avg Reward $\approx$ 600+
- **DDQN:** Avg Reward $\approx$ 350

DQN performed better likely due to optimistic overestimation aiding exploration.

IV. Discussion & Conclusion

This study demonstrates:

1) **Horizon Length Matters:** SARSA excels in small grids; MC dominates large, sparse environments.

Fig. 5: **Deep RL:** DQN (Blue) demonstrates superior convergence compared to Double DQN (Purple).

2) **Tabular Limitations:** Q-learning variants collapse in large state spaces.
3) **Exploration vs Bias:** Empirical results indicate that DQN performs better than Double DQN in this sparse-reward environment.

Work includes integrating Experience Replay and Prioritized Experience Replay for improved Deep RL efficiency.

REFERENCES

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

[2] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279-292, 1992.

[3] H. van Hasselt, "Double Q-learning," in *Advances in Neural Information Processing Systems*, vol. 23, 2010.

[4] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529-533, 2015.