ChaseRL: Comparative Analysis of TD Control

Q-Learning (Off-Policy) vs. SARSA (On-Policy)

Project Objective and MDP Design

The Goal: Balancing Risk and Reward

Train an agent to maximize target capture (+20.0 reward) while minimizing crashes (-20.0 penalty) in a dynamic, trap-filled environment.

Why Reinforcement Learning? The problem demands optimal, sequential decision-making under uncertainty. The policy must strategically balance long-term survival against short-term pursuit objectives.

Core Environment (MDP)

State Space (S): Discretized to 64 states based on 6 essential features (Danger Ahead, Prey Proximity, Prey Direction)

Action Space (A): Discrete [UP, DOWN, LEFT, RIGHT]

Reward Shaping: -20.0 penalty for crashes (Traps/Walls) enforces survival as highest priority

Algorithm Comparison: Off-Policy vs. On-Policy

Q-Learning (Off-Policy)

Mechanism: Learns the truly optimal action value *Q^**

Update Rule:

$$R + \gamma \max_{a'} Q(s', a')$$

Behavior: Optimistic—updates based on the best possible action from the next state, regardless of exploratory actions taken

SARSA (On-Policy)

Mechanism: Learns the value of the current exploratory policy *Q^\pi*

Update Rule:

$$R + \gamma Q(s', a')$$

Behavior: Conservative—factors in the risk of exploratory moves, leading to safer, risk-averse policies

Performance and Stability Analysis

Learning Curve: Q-Learning vs. SARSA (5000 Episodes)

Initial Convergence (0–1500 Episodes)

Both agents show rapid climb from -20 (crash penalty), confirming the effectiveness of reward shaping in acquiring survival instincts quickly.

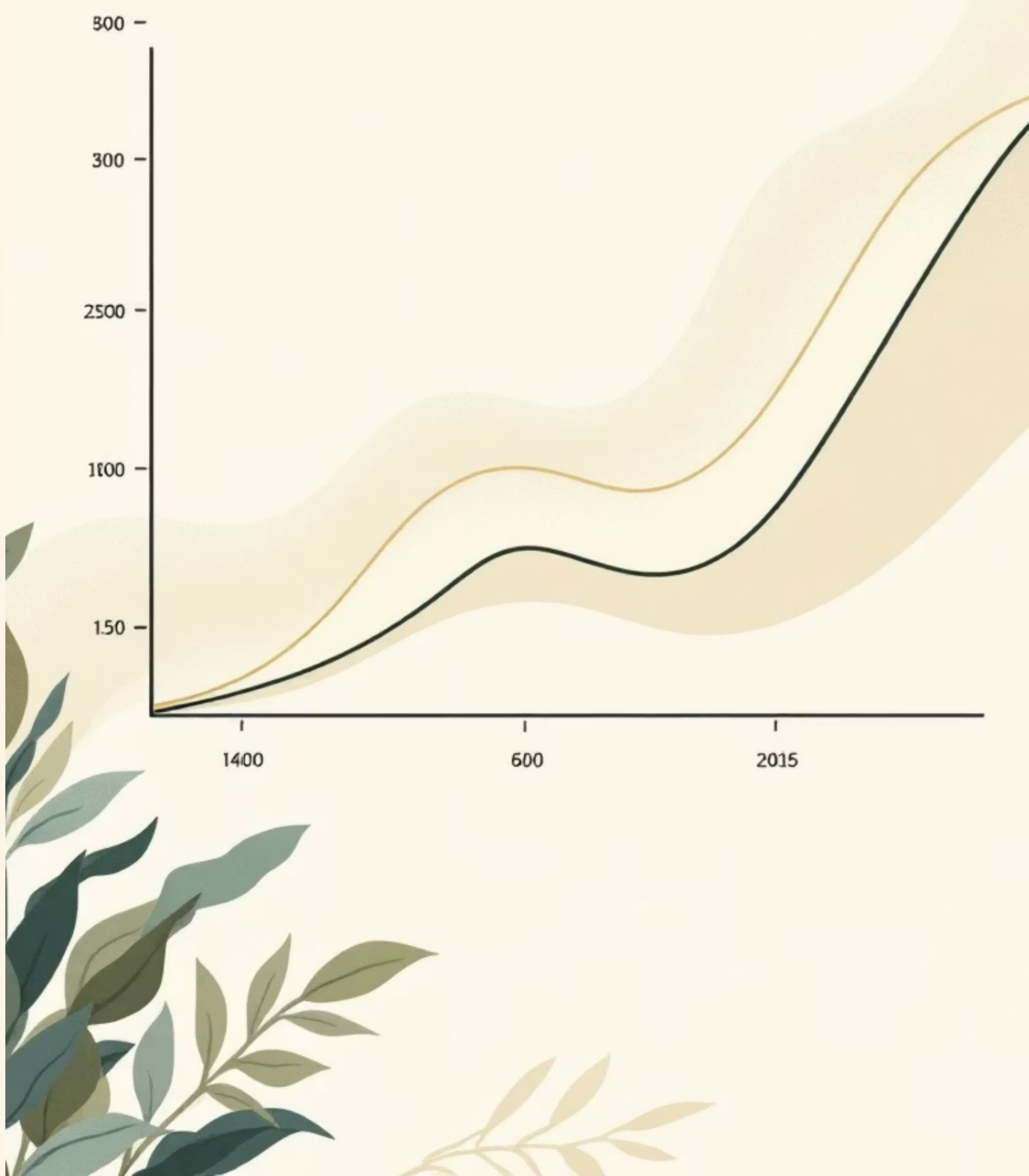
Off-Policy Peak Performance

Q-Learning (Blue) reaches higher, more aggressive reward peaks (spiking above +20), reflecting its inherent optimism and aggressive pursuit of optimal paths.

On-Policy Stability

SARSA (Orange) maintains smoother, more consistent average reward, demonstrating a stable policy that avoids high-risk zones Q-Learning explores.

Final volatility in both lines results from maintained exploration floor (εmin) and dynamic environment challenges.



Convergence Speed Comparison

Time to Solve: Average Reward > 5.0

1608	1632	24
Q-Learning Episodes	SARSA Episodes	Episode Difference
Off-policy approach	On-policy approach	Q-Learning advantage

Why Q-Learning is Faster

The update rule is inherently more efficient because it targets maximum reward directly. It doesn't waste time calculating the value of sub-optimal exploratory moves, accelerating overall convergence.

Why SARSA is Slower

Requires more total episodes because its cautious, iterative updates take longer for optimal value to propagate fully across the state space. Conservative evaluation slows convergence.

Policy Quality Trade-Offs

Demonstration Runs Analysis

Q-Learning Policy

High-Risk/High-Reward Strategy

- Lower average scores but highest peaks in learning curve
- Risks cutting corners close to traps
- Aggressive pursuit of theoretical optimality
- Best for environments where maximum performance justifies risk

SARSA Policy

Safer/More Robust Strategy

- Achieved consistent high scores (8-9 in demo runs)
- Maintains wider margin around traps
- Sacrifices theoretical optimality for reliability
- Essential for real-world applications where safety is paramount

Conclusion: The Fundamental Trade-Off

Q-Learning: The Optimistic Learner

Faster convergence and higher theoretical performance ceiling. Ideal when maximum reward justifies exploration risk and environment allows for aggressive optimization.

SARSA: The Conservative Learner

Safer, more stable, and essential for real-world applications where safety is paramount. Provides consistent, reliable performance with reduced variance.

Key Insight: This project validates the crucial trade-off between on-policy and off-policy TD control methods. The choice between Q-Learning and SARSA depends on whether your application prioritizes maximum performance (Q-Learning) or safety and reliability (SARSA).