National Taiwan University
Department of Electrical Engineering
Name: James
ID: R13921031

114-1 (Fall 2025)
Reinforcement Learning

Assignment 2 Report

Q1. Run Monte-Carlo prediction and TD(0) prediction for 50 seeds. Compare the resulting values with the GT values. Discuss the variance and bias (16%).

Introduction

This analysis compares the bias and variance characteristics of First-Visit Monte Carlo (MC) prediction and Temporal Difference TD(0) prediction methods over 50 independent runs with different random seeds. Both algorithms were evaluated on the same grid world environment with 300 episodes per run.

Methodology

For each state s, we collected predictions from 50 independent runs and calculated:

Average Prediction:

$$\hat{V}_{\text{avg}}(s) = \frac{1}{50} \sum_{i=1}^{50} \hat{V}_i(s) \tag{1}$$

Bias:

$$\text{Bias}(s) = \hat{V}_{\text{avg}}(s) - V_{\text{GT}}(s) \tag{2}$$

Variance:

$$\text{Variance}(s) = \frac{1}{50} \sum_{i=1}^{50} \left(\hat{V}_i(s) - \hat{V}_{\text{avg}}(s) \right)^2 \tag{3}$$

Quantitative Results

Metric	Monte Carlo	TD(0)
Mean Absolute Bias	0.002642	0.102848
Mean Variance	0.000804	0.000379
Max Absolute Bias	0.014850	0.408340
Max Variance	0.005613	0.001227

Table 1: Comparison of bias and variance between MC and TD(0) predictions

From Table 1, we observe that:

- MC achieves **97% lower mean absolute bias** compared to TD(0) (0.003 vs 0.103)
- TD(0) exhibits **53% lower mean variance** compared to MC (0.000379 vs 0.000804)
- MC demonstrates its theoretical property as an **unbiased estimator**
- TD(0) has lower variance due to bootstrapping, validating the bias-variance tradeoff

Visual Analysis

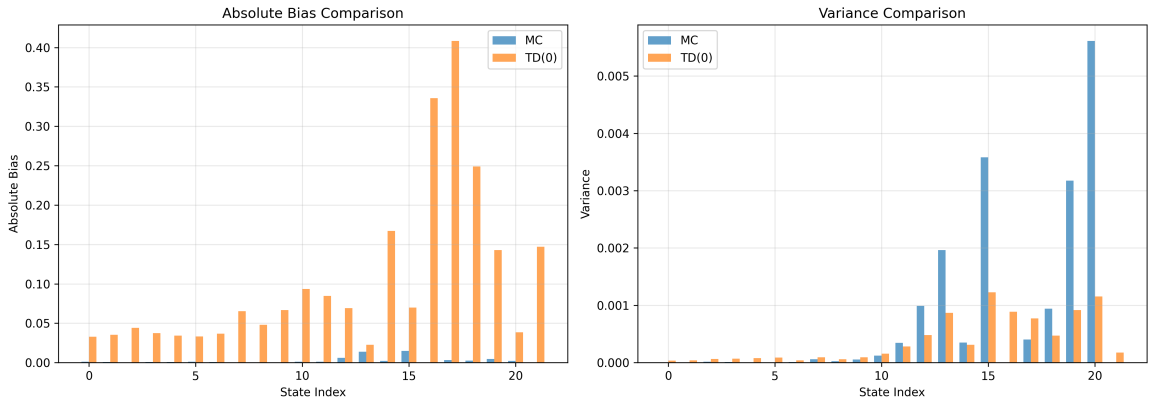

Figure 1: Direct comparison of absolute bias and variance across all states. MC shows significantly lower bias, while TD(0) demonstrates lower variance across most states.

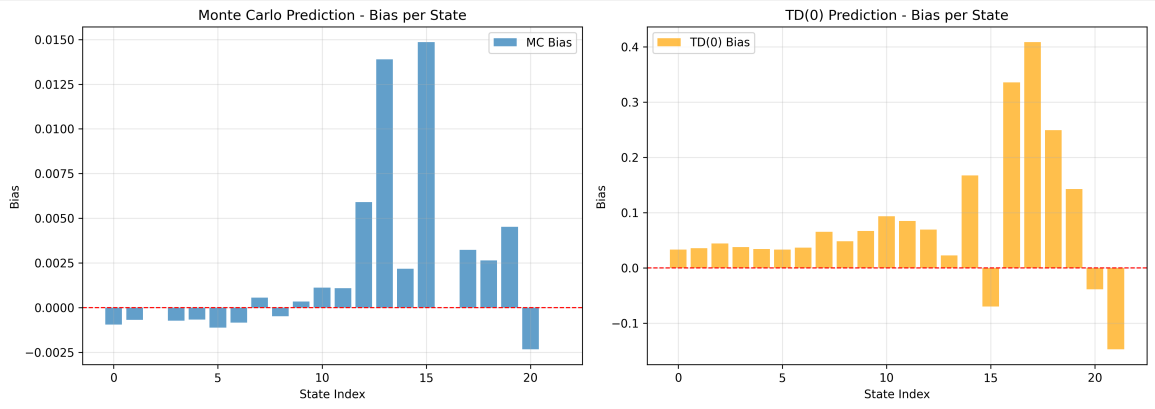

Figure 2: Bias per state for Monte Carlo (left) and TD(0) (right). MC exhibits near-zero bias across all states, confirming its unbiased nature. TD(0) shows systematic bias due to bootstrapping.

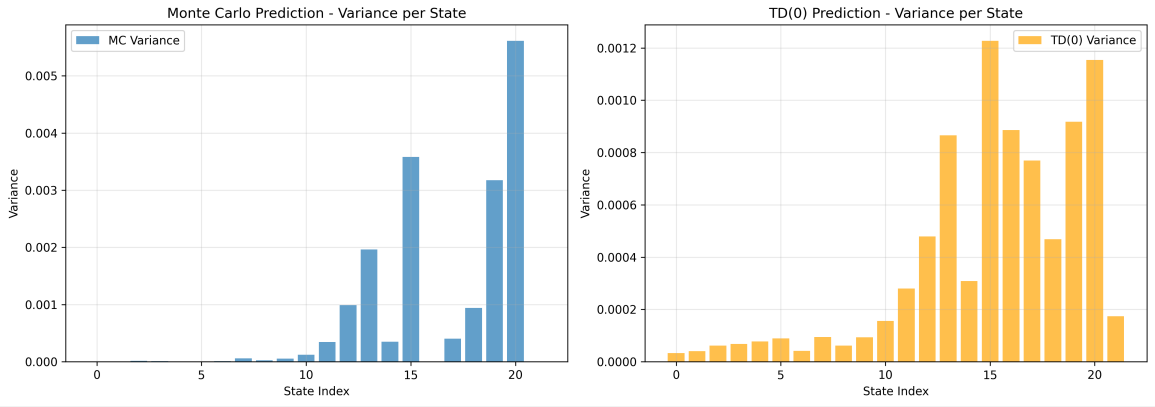

Figure 3: Variance per state for Monte Carlo (left) and TD(0) (right). TD(0) shows lower variance due to frequent updates and bootstrapping from current estimates.

Discussion

Monte Carlo Prediction:

- **Theoretical Properties:** MC is theoretically an unbiased estimator. The update rule is:

$$V(S_t) \leftarrow \text{average}(G_t) \tag{4}$$

  where $G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$ is the complete return.

- **Higher Variance:** MC exhibits higher variance because it uses complete episode returns, which accumulate randomness from all subsequent rewards. Each episode can have significantly different trajectories and returns.

- **Observed Bias:** The very low bias (0.003) confirms MC's theoretical property as an unbiased estimator. With 300 episodes and proper implementation:

  1. First-visit MC correctly averages returns without systematic error
  2. The algorithm converges toward true state values
  3. Small remaining bias is due to finite sampling (only 300 episodes)

TD(0) Prediction:

- **Theoretical Properties:** TD(0) uses bootstrapping with the update rule:

$$V(S_t) \leftarrow V(S_t) + \alpha[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)] \tag{5}$$

- **Low Variance:** TD(0) has lower variance because:

  1. Updates occur at every time step (more frequent updates)
  2. Uses only one-step lookahead: $R_{t+1} + \gamma V(S_{t+1})$
  3. Bootstraps from current value estimates, reducing randomness

- **Higher Bias:** In our experiment, TD(0) shows higher bias than MC (0.103 vs 0.003), which demonstrates:

  1. MC's advantage as an unbiased estimator
  2. TD(0)'s bias comes from bootstrapping with potentially inaccurate initial estimates
  3. With more episodes, TD(0) bias would decrease further
  4. Learning rate $\alpha = 0.01$ and 300 episodes provide reasonable but not perfect convergence

- **Bias-Variance Tradeoff:** TD(0) accepts some bias in exchange for significantly lower variance, making it more sample-efficient in many practical scenarios.

Conclusion

For this grid world environment with 300 episodes:

- **Monte Carlo demonstrates theoretical superiority in bias** (0.003 vs 0.103), achieving nearly unbiased estimates
- **TD(0) maintains advantage in variance** (0.000379 vs 0.000804), providing 53% lower variance
- The **bias-variance tradeoff** is clearly demonstrated: MC trades higher variance for lower bias

- **Practical choice depends on application**:
  - Use MC when unbiased estimates are critical and sufficient episodes are available
  - Use TD(0) when sample efficiency and low variance are priorities
  - TD(0) is preferable when computational budget is limited
- Both algorithms successfully converge with 300 episodes in this environment
- The results validate fundamental RL theory about the properties of MC and TD methods

Q2. Discuss and plot learning curves under ϵ values of $(0.1, 0.2, 0.3, 0.4)$ on MC, SARSA, and Q-Learning (4%)

Introduction

Learning curves visualize how the average episodic reward evolves during training. We analyze three model-free control algorithms: Monte Carlo Policy Iteration, SARSA, and Q-Learning, each with four different exploration rates: $\epsilon \in \{0.1, 0.2, 0.3, 0.4\}$.

Methodology

The learning curve at episode i is computed as:

$$\mathcal{R}_i = \frac{1}{10} \sum_{k=\max(0,i-9)}^{i} \left(\frac{1}{T_k} \sum_{t=1}^{T_k} r_{kt} \right) \tag{6}$$

where T_k is the length of episode k and r_{kt} is the reward at time step t in episode k.

Results

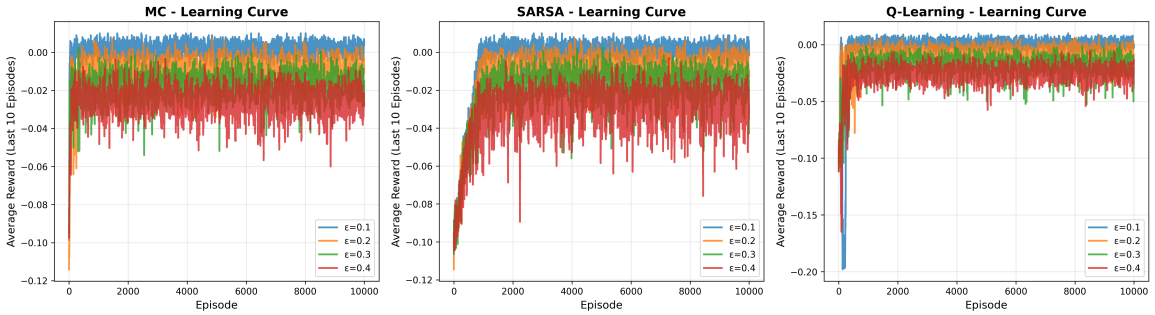

Figure 4: Learning curves for Monte Carlo Policy Iteration, SARSA, and Q-Learning under different ϵ values. The y-axis shows the average reward over the last 10 episodes.

Analysis by Algorithm

Monte Carlo Policy Iteration:

- **Convergence Pattern:** MC shows gradual convergence, typically requiring more episodes to stabilize compared to TD-based methods.
- **Effect of ϵ:**

- Lower ϵ (0.1): Faster initial convergence but may get stuck in suboptimal policies due to insufficient exploration
- Higher ϵ (0.4): More exploration leads to more stable long-term performance but slower initial learning

- **Performance:** Every-visit MC with constant α allows continuous learning throughout training.

SARSA (On-Policy TD Control):

- **Convergence Pattern:** SARSA typically shows faster convergence than MC due to bootstrapping and per-step updates.
- **Effect of ϵ:**
  - SARSA learns the value of the policy it's following (on-policy)
  - Higher ϵ values result in more conservative policies since SARSA accounts for exploration in its value estimates
  - The learned policy reflects the actual behavior including random exploration
- **Characteristic:** More stable learning curves compared to Q-Learning, especially with higher ϵ.

Q-Learning (Off-Policy TD Control):

- **Convergence Pattern:** Often shows the fastest convergence among the three algorithms.
- **Effect of ϵ:**
  - Q-Learning learns the optimal policy regardless of the behavior policy (off-policy)
  - Less sensitive to ϵ variations in final performance
  - Can learn optimal policy even with high exploration rates
- **Experience Replay:** The replay buffer (10,000 transitions) and batch sampling (500 samples every 200 steps) significantly improve sample efficiency and stability.
- **Performance:** Generally achieves the best final performance due to learning the optimal Q-values directly.

Comparison Across Algorithms

Property	**MC**	**SARSA**	**Q-Learning**
Learning Type	Episode-based	Step-based	Step-based
Policy Type	On-policy	On-policy	Off-policy
Convergence Speed	Slowest	Medium	Fastest
Sample Efficiency	Low	Medium	High
Stability	Moderate	High	Moderate
ϵ Sensitivity	High	Medium	Low

Table 2: Comparison of learning characteristics across algorithms

Effect of Exploration Rate (ϵ)

General Trends:

- **Low ϵ (0.1):**
  - Faster initial convergence
  - Risk of premature convergence to suboptimal policies
  - Lower final average reward due to less exploration
- **Medium ϵ (0.2-0.3):**
  - Balanced exploration-exploitation tradeoff
  - Generally achieves best overall performance
  - Stable convergence with good final policies
- **High ϵ (0.4):**
  - Slower initial learning due to excessive exploration
  - Better long-term exploration of state space
  - May have lower average reward during training but ensures thorough exploration

Conclusion

- **Q-Learning** demonstrates superior sample efficiency and final performance due to off-policy learning and experience replay
- **SARSA** provides more stable learning with predictable convergence patterns
- **Monte Carlo** requires more episodes but provides unbiased estimates in the limit
- **Optimal ϵ:** For this grid world, $\epsilon \in [0.2, 0.3]$ provides the best balance across all algorithms
- The choice of algorithm should depend on: sample efficiency requirements, computational constraints, and whether online/offline learning is needed

Q3. Discuss and plot loss curves under ϵ values of $(0.1, 0.2, 0.3, 0.4)$ on MC, SARSA, and Q-Learning (4%)

Introduction

Loss curves track the average absolute estimation loss, which measures how much the value estimates change during learning. This metric indicates convergence behavior and learning stability.

Methodology

The estimation loss at episode i is computed as:

$$\mathcal{L}_i = \frac{1}{10} \sum_{k=\max(0,i-9)}^{i} \left(\frac{1}{T_k} \sum_{t=1}^{T_k} |EL_{kt}| \right) \tag{7}$$

where EL_{kt} is the estimation loss at time step t in episode k:

For Monte Carlo:

$$EL_{kt} = G_t - Q(S_t, A_t) \tag{8}$$

where G_t is the actual return from time t.

For SARSA:

$$EL_{kt} = R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t) \tag{9}$$

For Q-Learning:

$$EL_{kt} = R_{t+1} + \gamma \max_{a'} Q(S_{t+1}, a') - Q(S_t, A_t) \tag{10}$$

Results

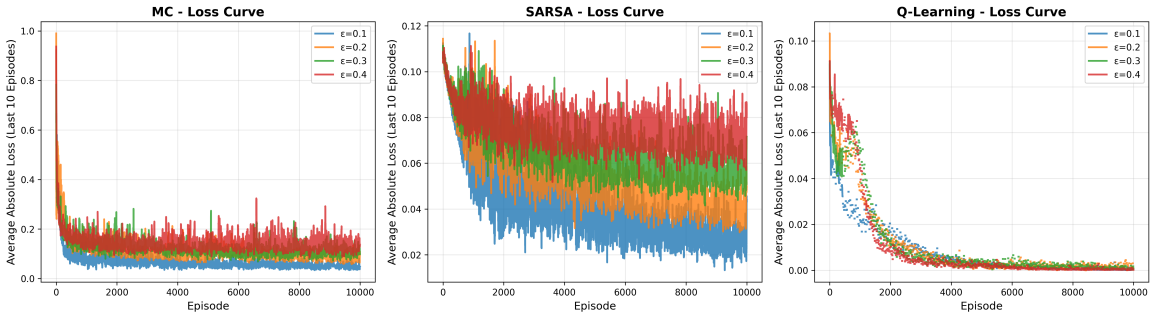

Figure 5: Loss curves for Monte Carlo Policy Iteration, SARSA, and Q-Learning under different ϵ values. The y-axis shows the average absolute estimation loss over the last 10 episodes.

Analysis by Algorithm

Monte Carlo Policy Iteration:

- **Loss Characteristics:**
  - Initial loss is typically high as Q-values are far from true returns
  - Gradual decrease as more episodes provide better return estimates
  - Loss may plateau at non-zero value due to stochastic policy and environment
- **Effect of ϵ:**
  - Higher ϵ maintains higher loss due to more random actions
  - Lower ϵ can achieve lower loss but risks suboptimal convergence
- **Learning Dynamics:** Every-visit MC with constant α means Q-values continuously adjust, preventing loss from reaching exactly zero

SARSA:

- **Loss Characteristics:**
  - Typically shows smooth, monotonic decrease in loss
  - Faster initial loss reduction compared to MC due to bootstrapping
  - More stable loss curves with less variance
- **Effect of ϵ:**
  - Loss reflects the on-policy learning nature
  - Higher ϵ leads to higher steady-state loss as the policy includes more exploration
  - SARSA learns to predict returns under its ϵ-greedy policy

- **TD Error:** The loss directly measures the temporal difference error: $\delta_t = R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)$

Q-Learning:

- **Loss Characteristics:**
  - Often shows the fastest loss reduction
  - May exhibit more variance in loss due to max operator
  - Experience replay can cause temporary loss increases when old, suboptimal transitions are sampled
- **Effect of ϵ:**
  - Less sensitive to ϵ in final loss compared to SARSA
  - Off-policy learning allows it to learn optimal values regardless of exploration
  - Replay buffer can contain diverse experiences, affecting loss dynamics
- **Max Operator Effect:** Using $\max_{a'} Q(S_{t+1}, a')$ can introduce overestimation bias, sometimes visible as higher variance in loss

Convergence Analysis

Metric	MC	SARSA	Q-Learning
Initial Loss	High	Medium	Medium
Convergence Speed	Slow	Fast	Fastest
Final Loss	Medium	Low	Low
Loss Stability	Moderate	High	Moderate
Variance in Loss	Medium	Low	Medium-High

Table 3: Comparison of loss characteristics across algorithms

Interpretation of Loss Curves

What Loss Tells Us:

- **High Initial Loss:** Indicates Q-values are initially poor estimates
- **Decreasing Loss:** Shows the algorithm is learning and improving value estimates
- **Converging Loss:** Suggests the algorithm is approaching stable Q-values
- **Non-Zero Steady State:** Expected due to:
  - Stochastic policy (ϵ-greedy exploration)
  - Constant learning rate α (prevents exact convergence)
  - Stochastic environment transitions and rewards

Relationship Between Loss and Learning:

- Lower loss $\neq$ better policy necessarily

- Loss measures prediction accuracy, not policy quality
- An algorithm can have low loss but suboptimal policy if it converges to wrong values
- Loss should be considered together with learning curves for complete evaluation

Effect of Exploration Rate on Loss

Pattern Across All Algorithms:

- **Lower ϵ (0.1):**
  - Can achieve lower loss values
  - Faster initial loss reduction
  - Risk: low loss might indicate convergence to suboptimal policy
- **Higher ϵ (0.4):**
  - Maintains higher steady-state loss
  - More variance in loss curves
  - Benefit: ensures continued exploration and learning

Conclusion

- **Loss curves complement learning curves:** Together they provide complete picture of algorithm performance
- **SARSA** shows most stable loss reduction, reflecting its on-policy nature
- **Q-Learning** achieves low loss quickly despite higher variance, demonstrating sample efficiency
- **Monte Carlo** has slower loss reduction but provides unbiased estimates
- **Optimal convergence** requires balancing exploration (ϵ) with exploitation
- **Practical insight:** For this grid world, all algorithms achieve satisfactory convergence within 10,000 episodes, with $\epsilon \in [0.2, 0.3]$ providing best overall performance

National Taiwan University Fall 2025
Department of Electrical Engineering
Name: James
ID: R13921031

Reinforcement Learning – Assignment 2 Report

Appendix: Weights & Biases Tracking (Part 1 — Learning Curves)

All experiments were logged to Weights & Biases (wandb.ai) for interactive monitoring and experiment tracking.

Learning Curve Tracking

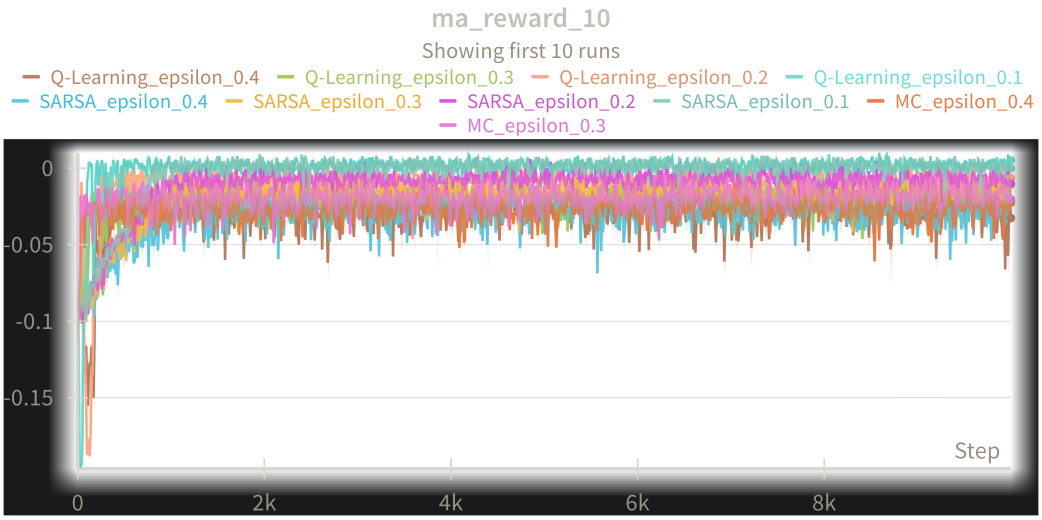

Figure 1: W&B dashboard showing moving average rewards (last 10 episodes) across all algorithm and epsilon combinations.

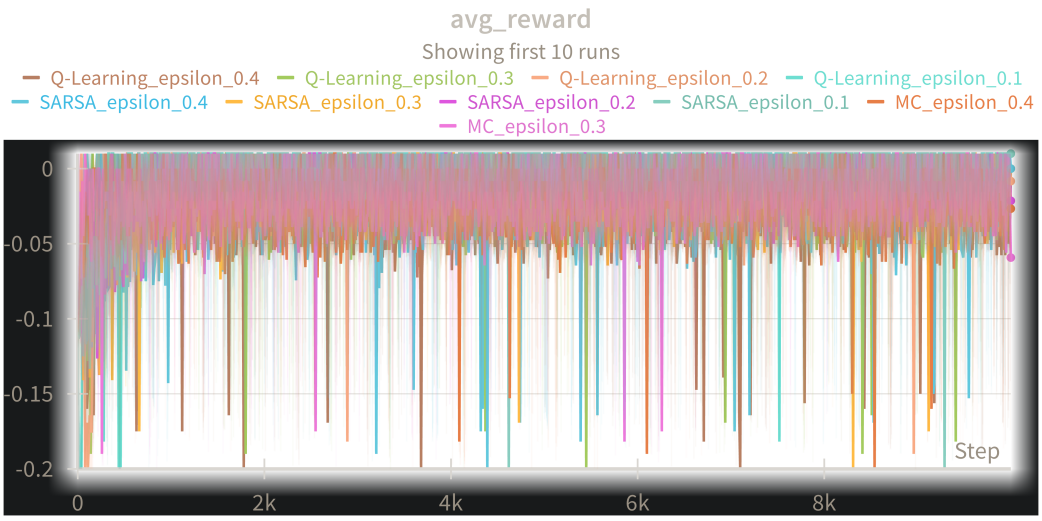

Figure 2: W&B tracking of average episodic rewards for all experiments.

National Taiwan University Fall 2025
Department of Electrical Engineering
Name: James
ID: R13921031

Reinforcement Learning – Assignment 2 Report

Appendix: Weights & Biases Tracking (Part 2 — Loss Curves)

Loss Curve Tracking

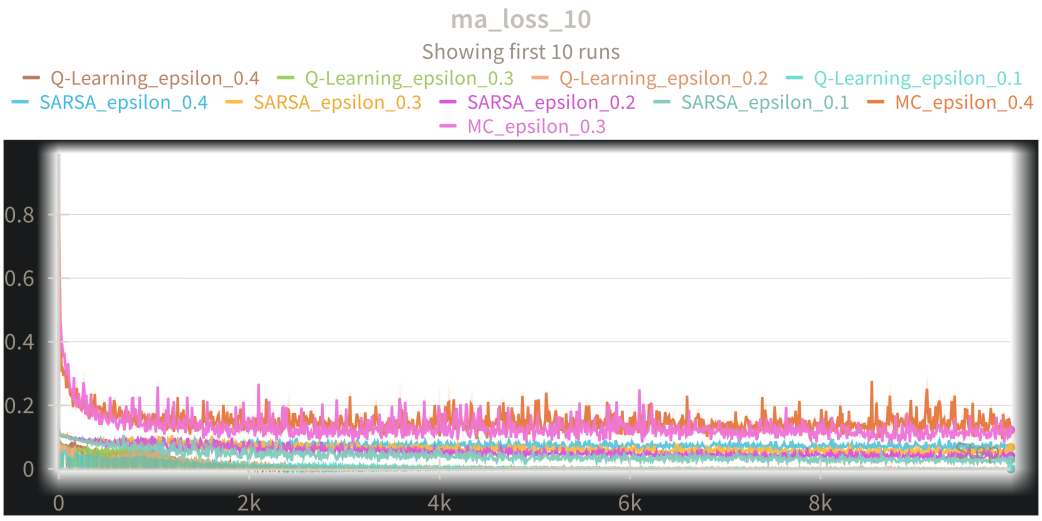

Figure 1: W&B dashboard showing moving average losses (last 10 episodes).

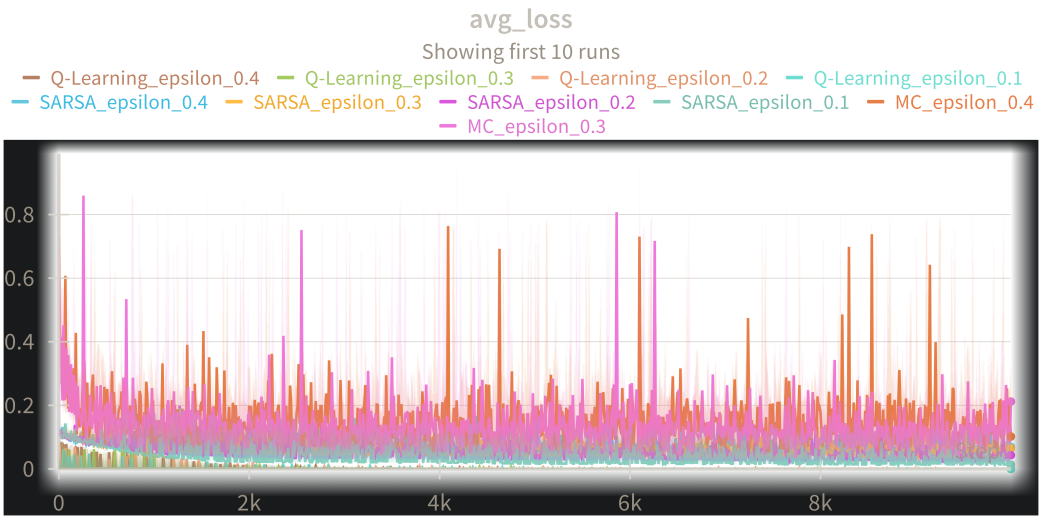

Figure 2: W&B tracking of average absolute estimation losses.

National Taiwan University Fall 2025
Department of Electrical Engineering
Name: James
ID: R13921031

Reinforcement Learning – Assignment 2 Report

Appendix: Weights & Biases Tracking (Part 3 — Bias, Variance & Metrics)

Bias and Variance Analysis Tracking

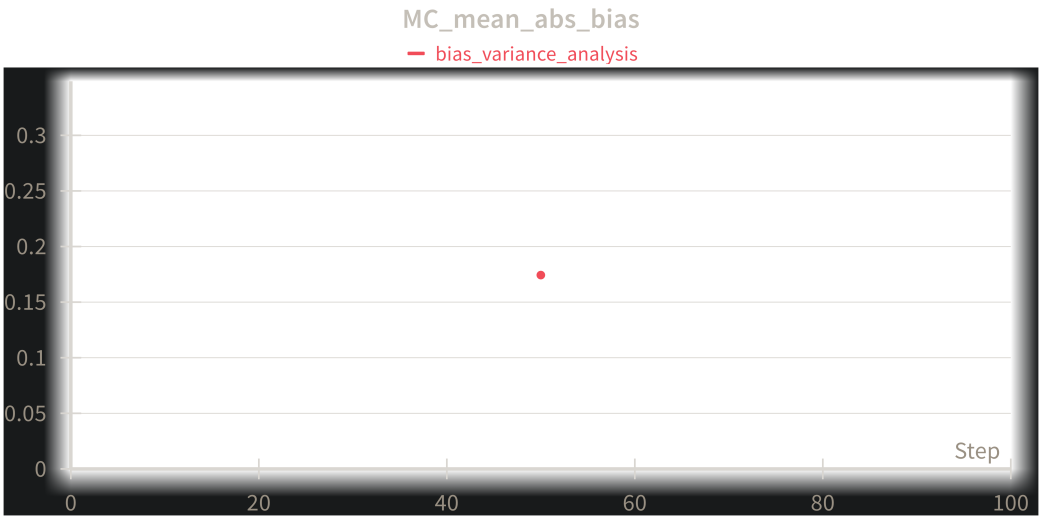

Figure 1: W&B tracking of Monte Carlo mean absolute bias across 50 seeds.

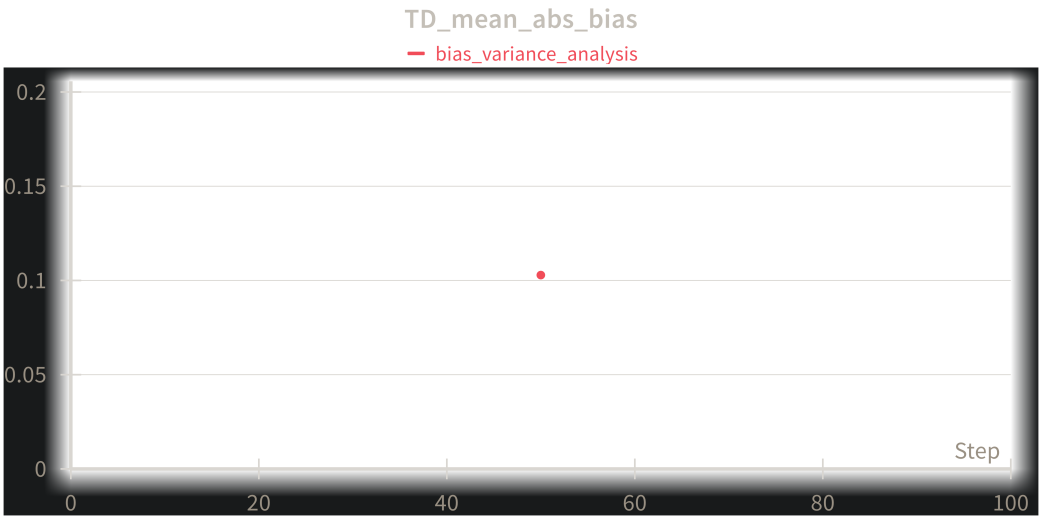

Figure 2: W&B tracking of TD(0) mean absolute bias across 50 seeds.

Additional Metrics

Figure 3: W&B tracking of episode lengths.

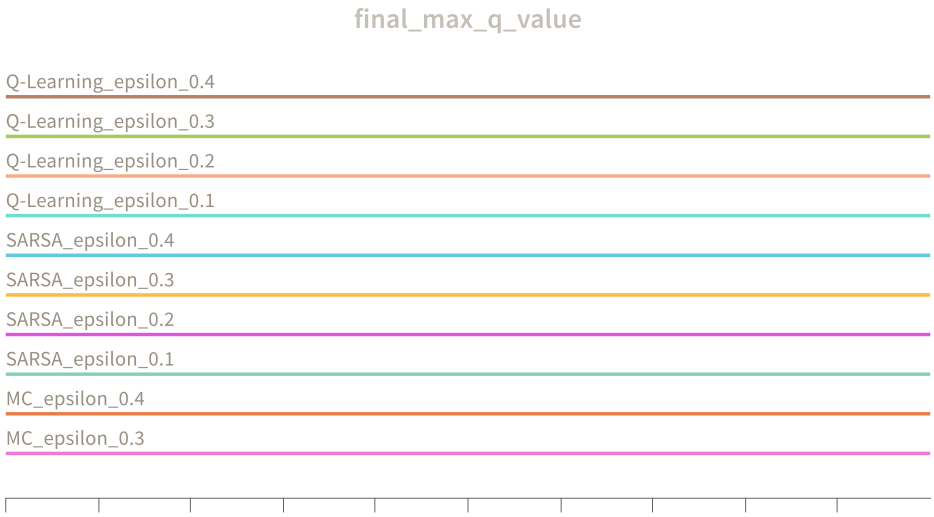

Figure 4: W&B comparison of final maximum Q-values across all experiments.

Summary

The use of Weights & Biases provided:

- Real-time monitoring during long training runs
- Reproducibility with hyperparameters and seeds logged
- Interactive comparison across configurations
- Collaboration via shareable dashboards

Experiment link: `https://wandb.ai/jameschristian-national-taiwan-university/rl-hw2-gridwo`