Rubric

Total Points: 100 + 5 Extra Credit

As a reminder, please give all or nothing for each point item. If a bullet is worth 3 points, award either 3 or 0 for that bullet. This keeps grading consistent across TAs.

Structured Feedback – CS7641 Fall 2025 RL Report

Continuing this semester, I want to put the point values by the section feedback. I have an example document I would like everyone to use as a template. You will still need to add individualized feedback for each student. Please reach out if you have any questions!

1. Requirements (10 points)

- **7 points total: DOCSTRING-GTusername reproducibility file**
  - **2 points:** Report written in LaTeX on Overleaf, and the project exists.
  - **2 points:** GitHub commit hash from GT Enterprise GitHub. Do not award if the hash/link is missing or non-specific (e.g., repo URL only).
  - **2 points:** Clear instructions to run code (environment, commands, and any package/init notes). If these are only on GitHub and not in the DOCSTRING, remove these 2 points.
  - **1 point:** DOCSTRING is a PDF.
- **1 point:** Report is 8 pages or fewer. Do not read beyond page 8; note to the student if over the limit. :contentReference[oaicite:5]index=5
- **1 point:** Graphs and text are legible at 100% zoom on Canvas. If 3+ figures have unreadable axes/legends at 100%, deduct this point.
- **1 point:** References are consistent (APA, MLA, or IEEE) and used correctly. Deduct if:
  - The style is mixed or misapplied (e.g., free-form URLs in body text instead of references).
  - There is no citation beyond course materials / environment docs. Students must connect to at least one external source. :contentReference[oaicite:6]index=6

2. Hypothesis (8 points)

Initial Hypothesis (4 points)

- **2 points:** States at least one clear, testable hypothesis about algorithm–environment behavior (e.g., "On CartPole, off-policy Q-Learning will reach longer average episode lengths than SARSA under the same ϵ-greedy schedule," or "PI will require fewer sweeps than VI on Blackjack but be more expensive per sweep.").
- **2 points:** Grounds the hypothesis in some prior evidence or theory: lecture content, Sutton & Barto, the FAQ, a paper, or well-known RL intuition (discounting, on/off-policy, exploration in stochastic vs near-deterministic domains, etc.). The support must be explicit, not just "this seems reasonable."

Follow-Through (4 points)

- **2 points:** Returns to the hypothesis in the discussion or conclusion, using the actual experiments to evaluate whether it held or not (even partially).
- **2 points:** Cites specific evidence from the report (plots, tables, metrics) in that follow-up. It is fine if the hypothesis is refuted; what matters is explicit, data-backed reflection rather than a throwaway line.

3. MDP Overview and Discretization (24 points)

- **Blackjack MDP (8 pts)**
  - (2 points) State tuple: (player sum, dealer up card, usable-ace flag) correctly described.
  - (2 points) Actions: hit / stick.
  - (2 points) Reward: +1 win, 0 draw, -1 loss (minor convention variants OK if clearly stated).
  - (2 points) Episodic nature explained: termination via bust or sticking, and the notion of an episode clearly articulated.
- **CartPole MDP (8 pts)**
  - (2 points) Lists the four continuous state variables (cart position, cart velocity, pole angle, pole angular velocity).
  - (2 points) Action space: left / right force.
  - (2 points) Reward: +1 per time step until failure.
  - (2 points) Termination conditions: angle and position thresholds (e.g., $\pm 12^\circ$ and ± 2.4), or clearly equivalent numeric ranges.
- **Comparison & Discretization Strategy (8 pts)**
  - (2 points) Highlights that Blackjack is small/discrete while CartPole is continuous and requires discretization for VI/PI and tabular RL.
  - (2 points) Explains the implication: tabular DP is straightforward for Blackjack, while CartPole needs a binned state space.
  - (2 points) Describes discretization approach (clamps, number of bins per feature, and how continuous states map to bins).
  - (2 points) Justifies bin choices: discusses fidelity vs runtime/memory trade-offs, ideally with order-of-magnitude state counts or simple numerical examples.

4. Value Iteration and Policy Iteration (30 points)

- **Explanation of VI and PI (8 pts)**
  - (2 points) VI: describes iterative Bellman optimality updates for all states until convergence.
  - (2 points) PI: describes the loop of policy evaluation (compute V^π) and greedy policy improvement.
  - (2 points) Notes convergence guarantees for finite MDPs with known transition dynamics.
  - (2 points) Clearly contrasts VI vs PI in terms of update style and computational trade-offs (e.g., more sweeps for VI vs heavier iterations for PI).
- **Convergence Visualizations (8 pts)** Four distinct plots: VI and PI on both Blackjack and CartPole.
  1. (2 points) VI – Blackjack: ΔV vs iteration (max or mean absolute change in V), showing a monotone or near-monotone decline below a stated threshold.
  2. (2 points) PI – Blackjack: number of policy changes per iteration (line or bar plot), with a rapid drop as the policy stabilizes.
  3. (2 points) VI – CartPole: ΔV vs iteration on the discretized state space; plot should mention discretization settings or reference them in the caption.
  4. (2 points) PI – CartPole: policy-change plot across iterations; ideally notes how discretization coarseness affects stability and convergence.
- **Convergence Analysis (7 pts)**
  - (2 points) Hyperparameter variation: explores at least θ and γ (3+ values ideal). If only 2 values for each are shown, award 1 point. If no hyperparameter variation for VI/PI, award 0.
  - (2 points) Explains why PI often converges in fewer iterations (due to greedy policy improvement), despite more expensive iterations.
  - (1 point) Reports iteration counts and explicit convergence criteria (e.g., $\max_s |V_{k+1}(s) - V_k(s)| < \delta$, or no policy changes).

- (2 points) Integrates quantitative results with theory; if results contradict expectations, discusses plausible causes (e.g., discretization choices, tolerance too loose, numerical issues).

- **Final Policy Comparison (4 pts)**
  - (2 points) Clearly states whether VI and PI produced the same policy (for Blackjack at minimum).
  - (1 point) If different, gives a concrete reason (e.g., different tie-breaking in arg max, rounding, or looser tolerance). If same, this is essentially a free point.
  - (1 point) Refers to a comparative visualization (heatmap, table, or policy map) in the text.

- **State Space Impact on CartPole (3 pts)**
  - (1 point) Explicitly acknowledges approximation error due to discretization.
  - (2 points) Discusses trade-offs quantitatively or with concrete arguments: more bins ⇒ larger state space, potentially better policies, but higher runtime/memory; fewer bins ⇒ faster but risk of aliasing and suboptimal policies. Being generous is fine if there's clear reasoning; hardware/time limits are real.

5. Model-Free Control with SARSA and Q-Learning (28 points)

Algorithm Explanation & On-/Off-Policy Nuance (4 pts)

- **4 points (Full):**
  - Correctly writes or clearly describes both updates:

$$SARSA: Q(s,a) \leftarrow Q(s,a) + \alpha\big[r + \gamma Q(s',a') - Q(s,a)\big]$$

$$Q-Learning: Q(s,a) \leftarrow Q(s,a) + \alpha\big[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\big]$$

  - Explicitly states that SARSA is on-policy (uses the action actually sampled under the current behavior policy) and Q-Learning is off-policy (targets $\max_{a'} Q(s',a')$ regardless of the behavior policy).
  - Explains at least one behavioral implication: SARSA tends to be more conservative / risk-averse under ϵ-greedy, while Q-Learning can over-commit to optimistic estimates.
- **2 points (Partial):** Correct equations and "on-policy vs off-policy" labels are present, but behavioral explanation is shallow or fuzzy.
- **1 point (Minimal):** Mentions both algorithms and that they are "different," but mixes the updates or mislabels on-/off-policy.
- **0:** Missing or fundamentally incorrect.

Learning & Convergence Diagnostics (8 pts)

Goal: show learning and stability for *both* algorithms on *both* environments, not just raw reward curves.

Minimum artifacts:

- At least **four** legible plots:
  - SARSA – Blackjack
  - Q-Learning – Blackjack
  - SARSA – CartPole
  - Q-Learning – CartPole
- Each environment must have:
  - A performance trajectory (reward or episode length vs episodes).
  - Some stability / convergence signal somewhere in the section: e.g., mean $|\Delta Q|$, TD-error decay, rolling variance/std of returns, or an explicit convergence criterion applied and discussed.

Scoring:

- **8 points:** All four (or more) plots present; every algorithm–environment pair has a performance curve; at least two stability-type diagnostics are shown and *interpreted*; each figure has at least one causal explanation (not just "curve goes up").
- **6 points:** Four curves across the two algorithms and environments, and at least one real stability diagnostic (or an explicit, applied convergence criterion) but coverage or interpretation is thin in one area.
- **4 points:** Curves present but almost entirely descriptive, or one algorithm–environment pair clearly missing.
- **2 points:** ≤ 2 plots, poor labeling, or only raw reward/length curves with no stability evidence and no convergence discussion.
- **0 points:** No meaningful learning plots.

If a student has only performance curves but *clearly states and uses* a convergence criterion (window + threshold) with discussion, you may go up to 6/8 instead of forcing 2/8.

Exploration Strategy Design & Impact (6 pts)

- **Full (6 points):**
  1. Describes at least one exploration scheme (e.g., ϵ-greedy with schedule or Boltzmann) with explicit parameters: initial ϵ, decay horizon/form, floor, temperature, etc. :contentReference[oaicite:11]index=11
  2. Justifies the schedule with respect to both environments (e.g., Blackjack needs persistent exploration of rare states; CartPole can safely reduce exploration once balancing is stable).
  3. Uses quantitative evidence to compare behavior under that schedule (episodes to reach a threshold return, variance after "convergence," etc.), and applies the same schedule when comparing SARSA vs Q-Learning to isolate on-/off-policy effects.
  4. Connects schedule mechanics to observed behaviors (e.g., "Fast decay on Blackjack caused under-exploration of usable-ace states; SARSA and Q-Learning both plateaued at lower returns.").
- **4 points:** Strategy and parameters are given, but rationale is superficial (e.g., "we used ϵ-greedy because it is standard") or evidence is very limited.
- **0 points:** No substantive exploration discussion.

Policy & Behavioral Analysis (6 pts)

- **Full (6 points):**
  1. Compares SARSA vs Q-Learning policies/behaviors on the environments. Uses heatmaps, value tables, or qualitative patterns (e.g., "Q-Learning more aggressively doubles down on high-variance states in Blackjack.").
  2. Relates these differences to on-/off-policy mechanics, exploration schedule, and variance (not just "they differ").
  3. Connects at least one RL policy (SARSA or Q-Learning) back to VI/PI policies: matches vs mismatches, with plausible causes (exploration residuals, sampling variance, discretization, tolerance).
- **4 points:** Some comparison is present but the causes are hand-wavy or purely descriptive.
- **2 points:** Differences are listed with essentially no causal reasoning.
- **0 points:** No policy comparison or incorrect explanations.

Robustness & Hyperparameter Sensitivity (4 pts)

- **4 points:**
  - Varies at least three hyperparameters per algorithm (e.g., α, γ, and an exploration parameter such as decay rate or floor), in line with the FAQ requirement that each model validate at least three hyperparameters. :contentReference[oaicite:12]index=12
  - Discusses robustness on *both* environments (qualitative OK for one if compute is tight).
  - States some convergence criterion (e.g., moving-average reward plateau, or mean $|\Delta Q|$ below δ) and uses it in the analysis.
  - Interprets stability vs speed trade-offs (e.g., high α faster initial gains but oscillation; lower γ shorter-sighted behavior in Blackjack).
- **3 points:** Clearly varies at least two hyperparameter families and touches both environments; third is weak or barely addressed.
- **1 point:** Only one hyperparameter explored or analysis is almost entirely qualitative.
- **0 points:** No real sensitivity analysis.

Extra Credit (Up to 5 points) – DQN / Rainbow Component on CartPole

Scoring is discrete: award **5**, **3**, **1**, or **0** only. No half-points.

Context: Extra credit focuses on implementing a DQN-style agent for CartPole (with replay + target network) and, ideally, at least one Rainbow-style component (Double DQN, Dueling, Prioritized Replay, Noisy Nets, n-step, or C51), and comparing it with tabular Q-Learning and SARSA.

5 / 5 (Full Credit) Working DQN with replay buffer and target network; at least one Rainbow component *or* a clearly articulated ablation. Architecture, key hyperparameters (buffer, batch size, learning rate, γ, ϵ schedule, target update) are described. Includes legible learning curves compared to SARSA/Q-Learning, discusses stability issues and mitigation, and provides quantitative comparisons (episodes to threshold, final mean±std reward, etc.) with causal explanations.

3 / 5 (Partial) DQN learns on CartPole with at least replay *or* a target network. Architecture/hyperparameters partially described. Some comparison to tabular methods is present but largely qualitative or thin on numbers/mechanisms. Rainbow-style modification may be present but shallowly discussed.

1 / 5 (Minimal / Legacy) Either:

- A serious but unsuccessful DQN attempt (architecture + hyperparameters + description of failure), or
- A tabular Q-Learning extension (e.g., Double Q-Learning) with some comparison to SARSA/DP, but no substantial DQN experiments.

0 / 5 (None) No substantive DQN or Rainbow-related implementation; only vague statements of intent.

Version Control

- 11/24/2025 - TJL Updated assignment to include Q-Learning from summer. Edited the point values to help with homogeny.