# Homework 2 Template

Use this template to record your answers for Homework 2. Add your answers using LaTeXand then save your document as a PDF to upload to Gradescope. You are required to use this template to submit your answers. **You should not alter this template in any way** other than to insert your solutions. You must submit all 8 pages of this template to Gradescope. Do not remove the instructions page(s). Altering this template or including your solutions outside of the provided boxes can result in your assignment being graded incorrectly. You may lose points if you do not follow these instructions.

Instructions to upload code have been provided in the handout.

## Instructions for Specific Problem Types

On this homework, you must fill in the blank for each problem; please make sure your final answer is fully included in the given space. **Do not change the size of the box provided.** For short answer questions you should **not** include your work in your solution. Only provide an explanation or proof if specifically asked. Otherwise, your assignment may not be graded correctly, and points may be deducted from your assignment.

**Fill in the blank:** What is the course number?

10-703

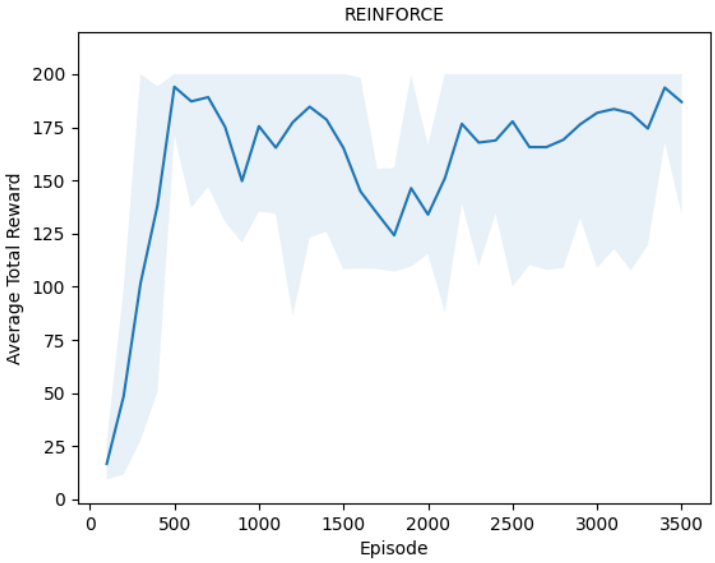
# Problem 0: Collaborators

Enter your team's names and Andrew IDs in the boxes below. If you do not do this, you may lose points on your assignment.

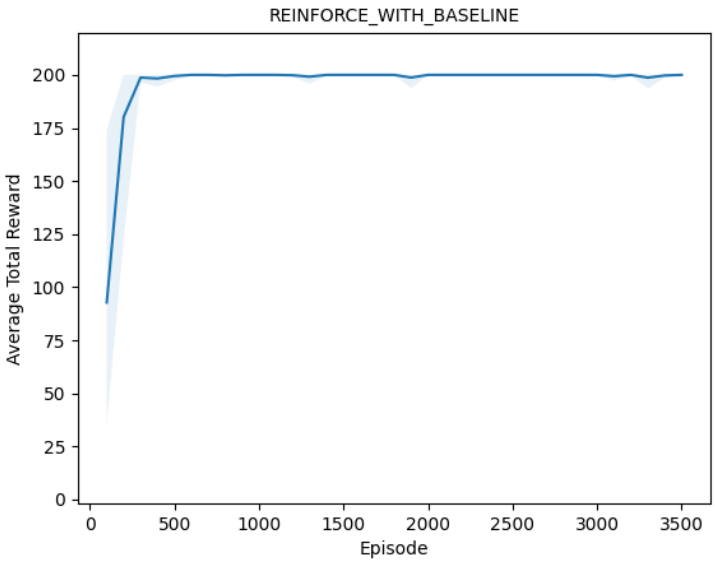
| | | | |
|---|---|---|---|
| Name 1: | Madhusha Goonesekera | Andrew ID 1: | mgoonese |
| Name 2: | Shrudhi Ramesh Shanthi | Andrew ID 2: | srameshs |
| Name 3: | Siddharth Ghodasara | Andrew ID 3: | sghodasa |

# Problem 1: REINFORCE (48 pts)

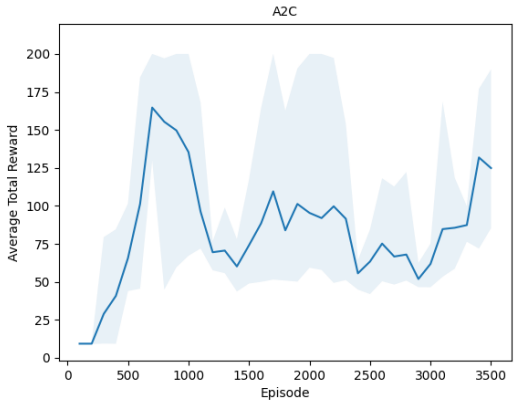
## 1.1 Reinforce plot (10 pts)

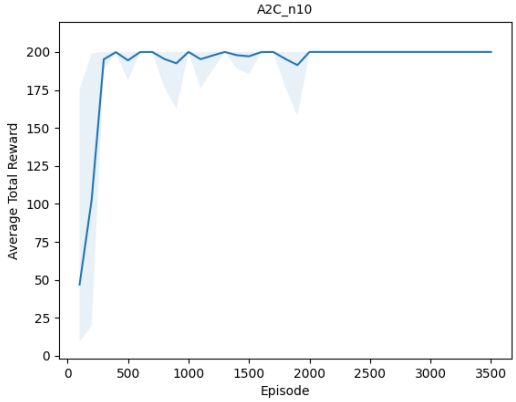


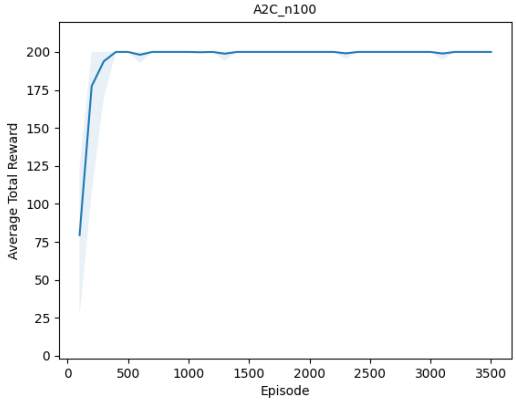

## 1.2 Reinforce with baseline plot (10 pts)



## 1.3 N-step A2C (20 pts)




In order n=1, n=10, n=100

## 1.4 N-step A2C & REINFORCE with baseline (4 pts)

N-Step A2C, REINFORCE, and REINFORCE w/ Baseline are all policy gradient methods. REINFORCE and REINFORCE with baseline are more like Monte Carlo because the policy is updated from complete episodes, while N-Step A2C is like N-Step TD.

As we've seen before with Monte Carlo and TD, the two are equivalent where the n-step is the same length as the episode. By the same principle, N-Step A2C and REINFORCE will be equivalent when the N-Step matches the length of the episode. From a math/proof standpoint this can be represented as $N \to \infty$.

REINFORCE with Baseline is a little different, given that the return is subtracted by some baseline function. But we can see a strong similarity in the update for A2C and REINFORCE w/ Baseline, notably the portion: $G_t - b_\omega(S_t)$ for REINFORCE w/ Baseline and $G_t - V_\omega(S_t)$ for A2C. The condition for equivalence is where $b_\omega(S_t) = V_\omega(S_t)$ and where $N \to \infty$

**TL;DR:**
**A2C = REINFORCE when $\mathbf{N \to \infty}$, in other words when the steps is the same as episode length**
**A2C = REINFORCE w/ Baseline when $\mathbf{b_\omega(S_t) = V_\omega(S_t)}$ & $\mathbf{N \to \infty}$**

## 1.5 REINFORCE with & without baseline (4 pts)

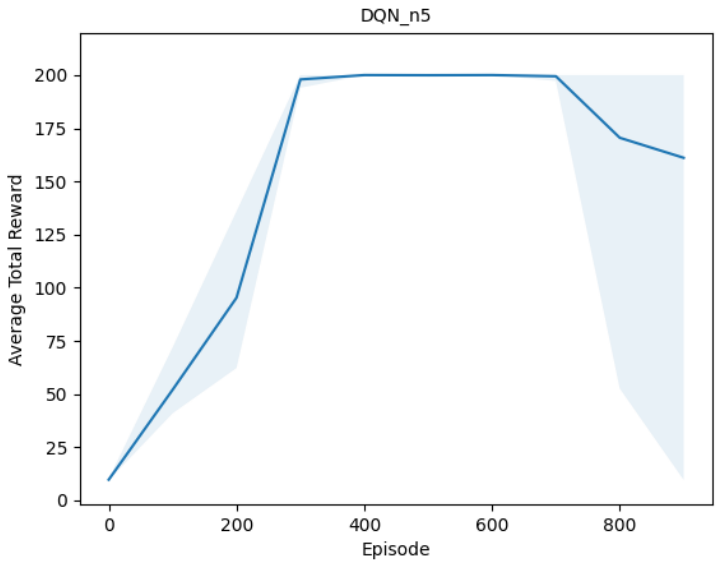
Yes, adding a baseline does improve performance. The big reason for this is because REINFORCE works in a purely additive way. That is, actions are not punished, just rewarded less. But the baseline in REINFORCE with baseline allows the algorithm to effectively punish bad states because states can now receive negative values. This allows for faster convergence and ultimately few episodes to reach that convergence to optimal behavior

# Problem 2: DQN (32 pts)

## 2.1: Understanding TD & MC (6 pts)

1. **False**: The big differentiator between TD and Monte Carlo is the fact that TD doesn't need the whole trajectory, and is able to update estimates from the current state.
2. **False**: Monte Carlo (MC) rollouts typically rely on complete episodes to calculate the return (the sum of rewards from a starting point to the end of the episode) because they update value estimates based on the actual outcomes of full trajectories. For non-terminating episodes, there is no natural "end" from which to calculate the return, which is crucial for traditional MC methods.Atleast from our class discussions on conventional MC rollouts this seems to be the case. In theory, we could modify MC to incorporate truncated rollouts or incremental rollouts to avoid this problem.

## 2.2 Average Cumulative Reward Plot (15 pts)




Note: Title has artifact "n5" leftover from PG title printing

## 2.3 DQN vs Policy gradient algorithms (5 pts)

There are a few reasons why DQN performs better than policy gradients. In terms of convergence, DQN does better than RwB and A2C in convergence rate, and much better in convergence and performance to REINFORCE. This can be attributed to a couple of key things: (1) DQN Performs better for discrete action spaces while Policy Gradients do better for continuous action spaces. (2) DQN outperforms policy gradient algorithms on tasks like CartPole-v1 due to its stability, sample efficiency, lower variance in learning, and effective exploration-exploitation balance.

## 2.4 Pros and Cons of Policy Gradient Methods (6 pts)

Pros of Policy Gradient (PG) Methods:

- High Dimensional & Continuous action spaces:
  - PG methods, such as N-Step A2C, can handle continuous action spaces
  - This makes them ideal for tasks such as robotic control or other environments that require smooth, non-discrete actions
  - In addition to the above, they scale well with large or complex action spaces, where methods like DQN struggle due to the discrete nature of the Q-value functions
- On-Policy Learning
  - PGs work well in environments where the current policy's actions must be used to gather data which ensures that exploration strategies are tightly aligned with the current learning process
- Direct Policy Optimization
  - By doing direct policy optimization, policy gradient methods can adapt efficiently to the task at hand
  - Particularly in environments with complex or dynamic reward structures

  Cons of Policy Gradient (PG) Methods:
  - Sample Inefficiency
    - PG methods often require a large amount of environment interactions to learn effectively
    - This is due to the fact that they can't use past experiences like DQN can with experience replay
  - Higher Variance
    - The direct optimization of the policy can lead to high variance in the gradient estimates, which can make convergence slower and less stable compared to Q-value based methods like DQN
  - Local Optima susceptibility
    - Since PG directly optimizes the expected return, they are more prone to get stuck in local optima, especially in complex environments

# Feedback

**Feedback**: You can help the course staff improve the course for future semesters by providing feedback. You will receive a point of you provide actionable feedback. What was the most confusing part of this homework, and what would have made it less confusing?

Perhaps the most confusing part of this assignment was navigating the provided functions and determining the limits of what we can implement on our own vs what was provided. More specifically, for the provided functions there were several instances where certain inputs either were not needed, or had no reasonable explanation what their purpose was. Moving forward, it would be helpful to include a function header explaining what the inputs are for and/or their expected type.

**Collaboration**: Detail the work division amongst your group below.

The core understanding of the deliverables and outline was done collaboratively as a group. **Question-1:** The baseline implementation and outline was carried out by Madhusha. The implementation was carried out by the entire team. The debugging and refinement was carried out by Madhusha and Siddharth. **Question-2:** The baseline implementation and outline was carried out by Shrudhi and Siddharth. The implementation as carried out by the entire team. The debugging and refinement was carried out by Shrudhi and Siddharth.

**Time Spent**: How many hours did you spend working on this assignment? Your answer will not affect your grade. Please average your answer over all the members of your team.

| | |
|---|---|
| Alone | 15 |
| With teammates | 60 |
| With other classmates | 0 |
| At office hours | 0 |