Homework 4 Submission

Use this template to record your answers for Homework 4. 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 9 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 should also export your code as a .py file and upload it to the **separate** Gradescope coding assignment. Remember to mark all teammates on **both** assignment uploads through Gradescope.

Instructions for Specific Problem Types

On this homework, you must fill in blanks 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.

Fill in the blank: What is the course number?

10-703

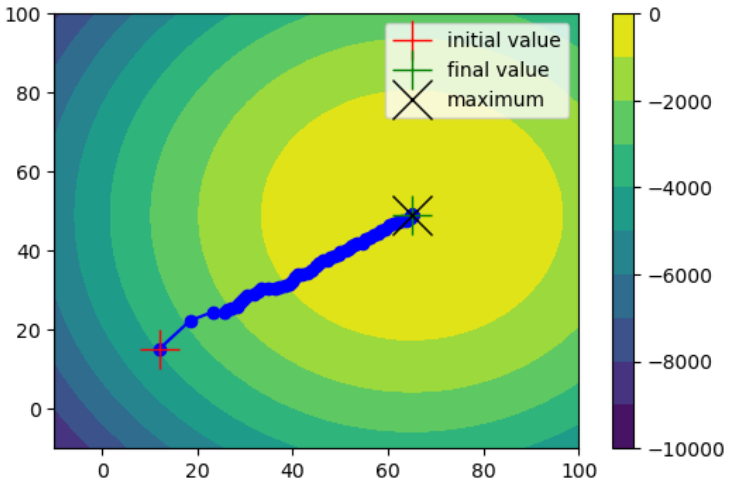
Problem 0: Collaborators

Enter your team members' names and Andrew IDs in the boxes below. If you worked in a team with fewer than three people, leave the extra boxes blank.

Name 1:	Mike Anoruo	Andrew ID 1:	manoruo
Name 2:		Andrew ID 2:	
Name 3:		Andrew ID 3:	

Problem 1: CMA-ES (24 pts)

1.1 Plot of CMA-ES on simple objective function (10 pts)



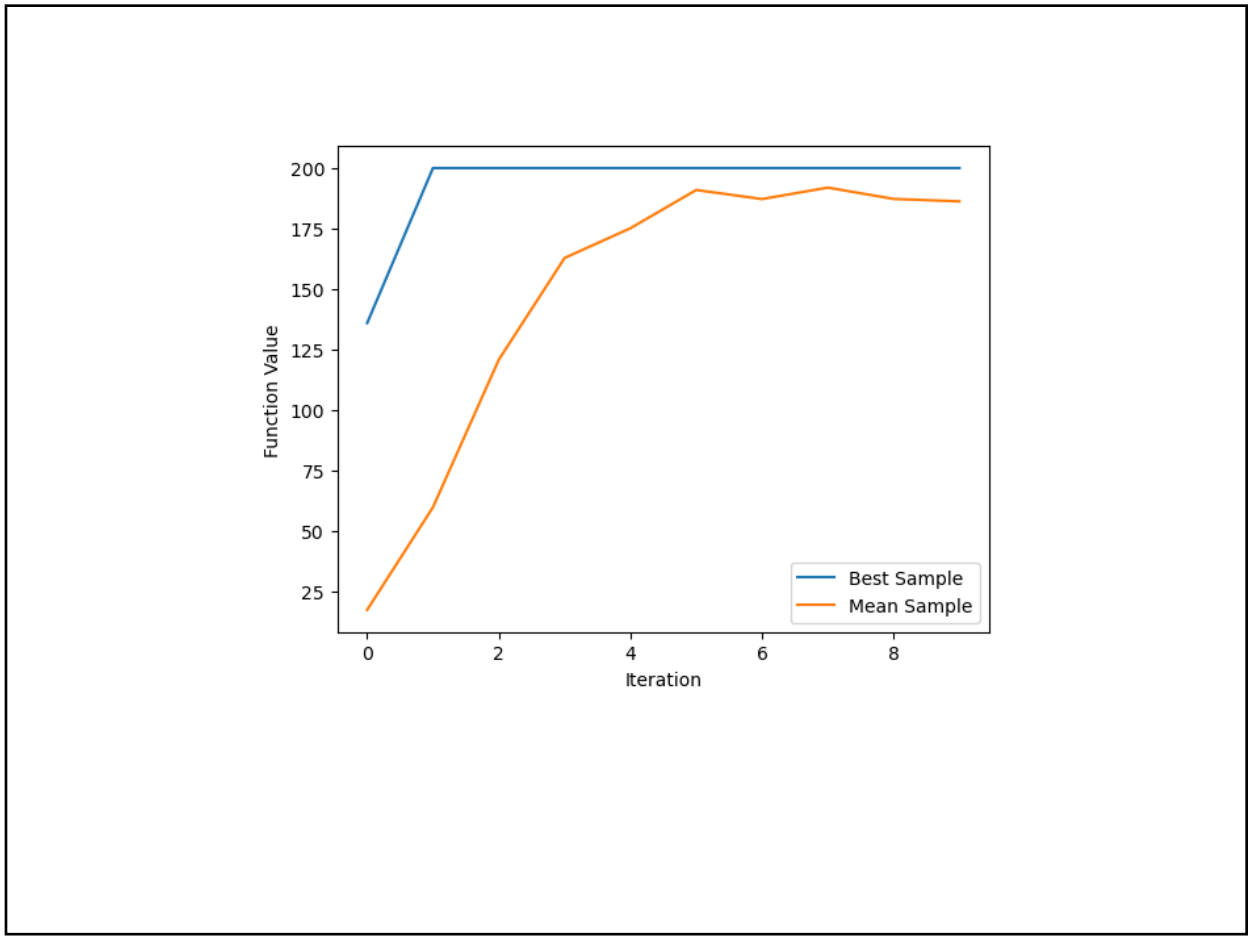
1.2 RL reward of fixed policies (4 pts)

$x = (-1, -1, -1, -1, -1)$: 15.6

$x = (1, 0, 1, 0, 1)$: 14.4

$x = (0, 1, 2, 3, 4)$: 9.4

1.3 Plot of CMA-ES on Cartpole (10 pts)

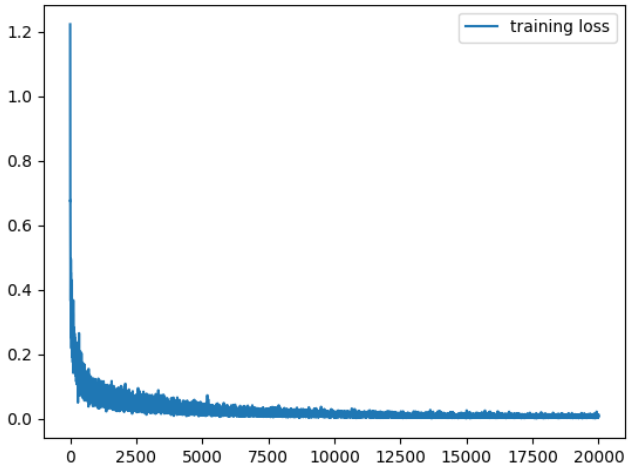


Problem 2: Imitation Learning (62 pts)

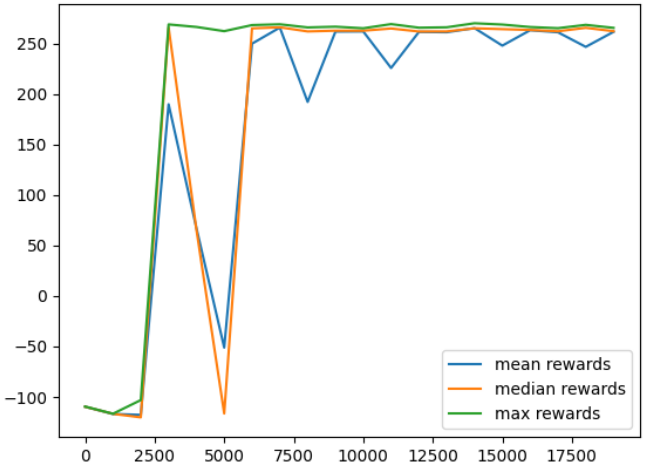
Problem 2.1: BC (14 pts)

2.1.1 Loss Plot + Final Loss Value BC (6 pts)

Final loss: loss 0.011179



2.1.2 Rewards Plot BC (6 pts)



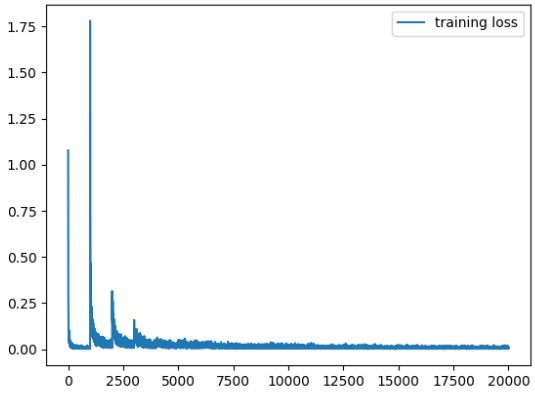
2.1.3 GIF link BC (2 pts)

You can find this in my submission at: visuals/gifs_BC.gif

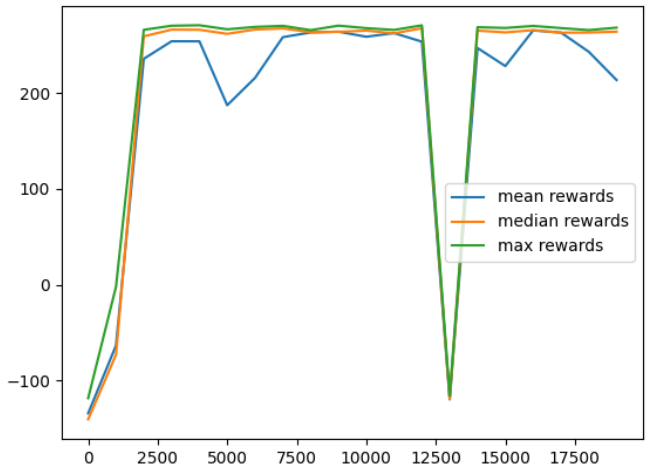
Problem 2.2: DAgger (18 pts)

2.2.1 Loss Plot DAgger (6 pts)

Final Loss: loss 0.005250



2.2.2 Rewards Plot DAgger (6 pts)



2.2.3 GIF link DAgger (2 pts)

You can find this in my submission at: visuals/gifs_DAgger.gif

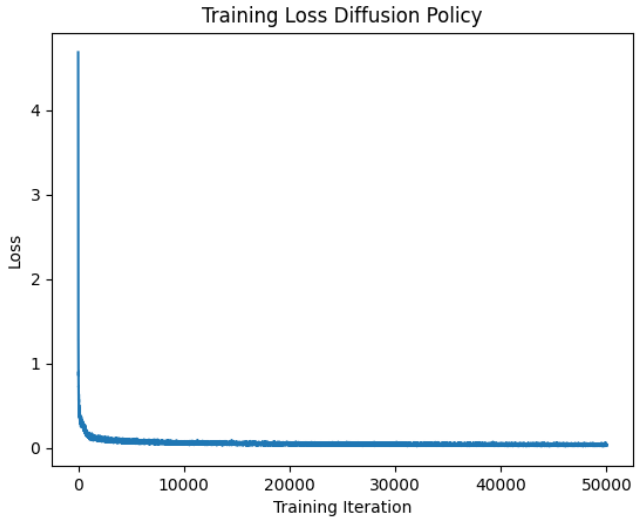
2.2.4 Compare DAgger training with BC (written, 4 pts)

DAgger worked better than BC because it continually queries the expert on the states the policy visits, correcting errors and reducing compounding mistakes, whereas BC only learns from the expert's demonstration distribution and suffers from distribution shift.

Problem 2.3: Diffusion Policy (30 pts)

2.3.1 Loss Plot + Final Loss value Diffusion Policy (6 pts)

Final loss = 0.03161560744047165



2.3.2 Rewards Diffusion Policy (15 pts)

2.3.2.1; 3 actions evaluated in a row (5 pts)

avg trajectory time: 845.33 mean: 201.87 median: 259.39 max: 266.60

2.3.2.2; 2 actions evaluated in a row (5 pts)

avg trajectory time: 806.03 mean: 183.63 median: 258.10 max: 266.30

2.3.2.3; 1 action evaluated in a row (5 pts)

avg trajectory time: 865.83 mean: 208.96 median: 259.27 max: 264.89

2.3.3 GIF link Diffusion Policy (2 pts)

You can find this in my submission at: visuals/gifs_diffusion.gif

2.3.4 Compare diffusion policy and simple model runtime (written, 4 pts)

The diffusion policy is slower because it generates each action through an iterative denoising process over many timesteps, whereas BC or DAgger output actions directly in a single forward pass.

2.3.5 Compare diffusion policy with different actions in a row runtime (written, 3 pts)

Predicting more actions at once reduces the number of diffusion model calls per trajectory, so trajectories with higher num_actions_to_eval_in_a_row run faster on average.