Homework 4 Template

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

Problem 1: Model-Based Reinforcement Learning with PETS (100 pt)

1.1 Model-based Predictive Control (25 pts)

1.1.1 CEM (without MPC) with ground-truth dynamics (10 pt)

Success percentage | 86% |

1.1.2 Random sampling with ground-truth dynamics. (10 pt)

Success percentage without MPC | 68% |

Success percentage with MPC | 94% |

How does the performance of random sampling performance compare to that of CEM?

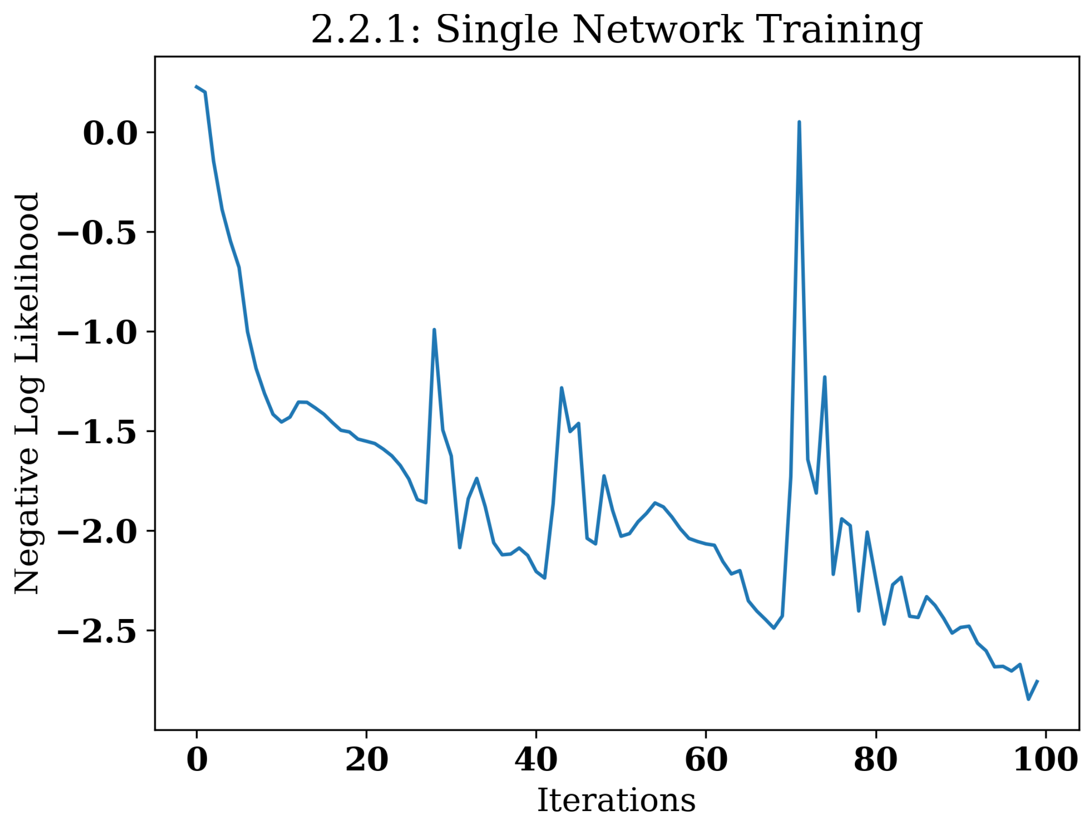
When comparing CEM without MPC to Random Sampling without MPC, we can see above that CEM performs better. There is close to a 18 percent improvement from Random Sampling to CEM. This does make sense, as CEM does fitting of the data to iteratively improve the final set, while random sampling is, well, random sampling. When involving MPC, we can see that random sampling now outperforms CEM without MPC. But, this improvement only highlights the power of MPC, and doesn't really make a suitable comparison between CEM and Random Sampling Directly.

1.1.3 MPC vs. open-loop control (10 pt)

MPC outperforms open-loop control in dynamic, uncertain environments and tasks with constraints due to its ability to predict future states, integrate real-time feedback, and handle constraints explicitly. This makes it ideal for scenarios like autonomous driving or robotic manipulation in constrained spaces. In contrast, open-loop control is better suited for static, predictable environments where disturbances are minimal and computational resources are limited, such as simple manufacturing tasks. The key advantages of MPC include robustness to disturbances, adaptability, and constraint handling, but it requires significant computational resources and depends heavily on an accurate model. Open-loop control is computationally simpler and easier to implement but lacks the adaptability and feedback mechanisms needed for complex, real-world environments. Therefore, the choice between them depends on the trade-off between computational feasibility and the complexity of the task.

1.2 Single probabilistic network (40 pts)

1.2.1 Training loss plot (10 pt)



1.2.2 MPC with random sampling (12 pt)

Success percentage 32%

1.2.3 MPC with CEM (12 pt)

Success percentage 40%

1.2.4 Random sampling vs. CEM (6 pt)

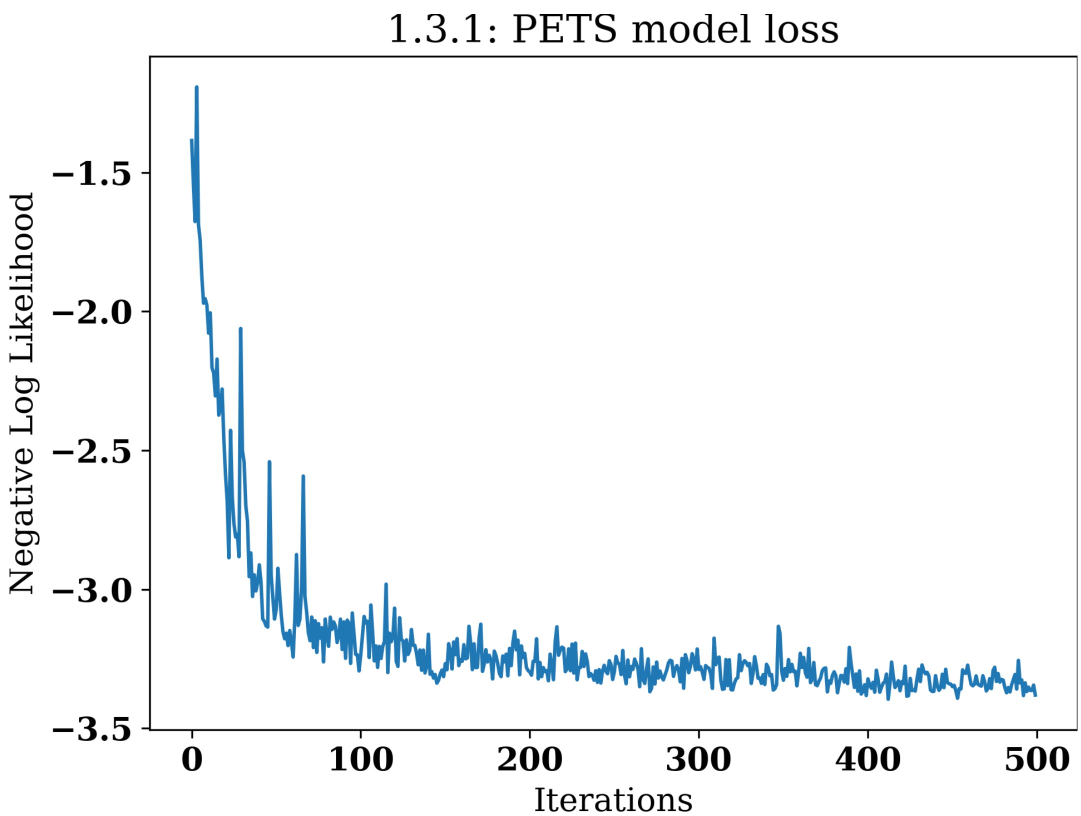
In the results above, CEM performed better than Random Sampling (by 8 percent), though not to the same level as when we were using ground-truth dynamics. CEM uses iterative refinement, updating the model of the distribution based on high-performing samples. Its more efficient and effective for complex problems, but its computationally more expensive. The choice between random sampling and CEM for planning depends on the complexity of the task and the problem's characteristics.

In this case, MPC did perform well generally, but again we can see that compared to ground-truth dynamics MPC suffered much more using the model. This highlights how MPC is dependent on the quality and generalization of the model.

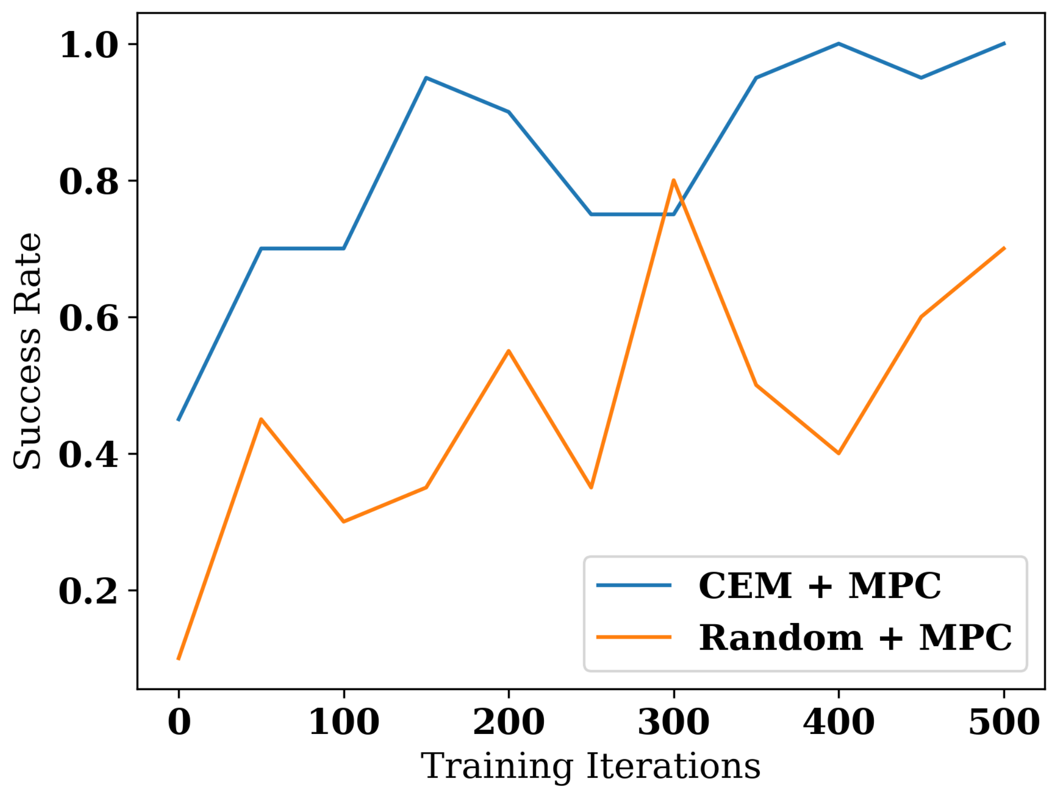
The derived policy succeeds more often when tested with CEM, achieving a higher average success rate (40%) and better rewards due to its optimized action selection. In contrast, it fails more frequently with the random planning method, which lacks strategic guidance, resulting in a lower success rate (32%) and suboptimal rewards.

1.3 MBRL with PETS (35 pts)

1.3.1 Training loss plot (10 pt)



1.3.2 Test percentage of successes plot (10 pt)



1.3.3 Limitations of MBRL (5 pt)

VERIFY THIS BEFORE SUBMISSION Model-Based Reinforcement Learning (MBRL) has several limitations, including reliance on an accurate system model, which can be difficult to obtain in complex or dynamic environments. MBRL also tends to be computationally expensive due to the need to frequently simulate and optimize over the learned model. Moreover, it may struggle in highly stochastic environments or with noisy data, as small errors in the model can lead to significant performance degradation. However, MBRL is preferred in scenarios where data efficiency is crucial, as it leverages a learned model to simulate future states and actions, reducing the need for extensive real-world interactions. In contrast, policy gradient methods, while more robust to model inaccuracies, typically require more data and are less efficient in environments where a good model can significantly reduce the amount of real-world exploration needed. Thus, MBRL is more suitable when system dynamics are well-understood and computational resources are available, while policy gradients excel in environments with complex dynamics or high uncertainty.