

# DQN and DDPS in Loma

Ziyuan Lin

2025.5.18

## 1 General Plan

In this final project, I will use Loma to implement a specific neural network: Deep Q-network. The implementation will include both the forward pass and backward pass. If time permitted, I will also implement DDPS in Loma.

## 2 Detailed Plan

The task is divided into 4 parts: Data collection and preprocessing. Core deep Q-network. The loss function  $J(\theta)$ . The decision making and state update. Only the second part and third part will be written in Loma since they are the only functions that need differential. For the first and last part, I will use python/C++ wrapper. For the dataset, I will use one of Atari 2600 games. If possible, I will strictly follow the method in [1]. The usage of Loma will be entirely in training procedure. If time permitted, I will implement DDPS, which I will take a similar strategy as DQN.

## 3 Potential Difficulty in the Project

The first difficulty is the interface problem. As I will pick C++ as my wrapper, I need to be careful about the input to the C code generated by Loma and also the output from it. The second difficulty will be the matrix implementation. The third potential difficulty is the performance issue. If possible, I will leverage GPU for matrix operation. Finally, if I further implement DDPS, then learning is necessary. But I still believe it to be super interesting to study how differentiation could be useful for DPS.

## 4 Minimum Viable Product(MVP)

At the minimum scale, I will implement the DQN in Loma. I will write the report with training curve, gpu utilization, final result. I will also give the structure of DQN.