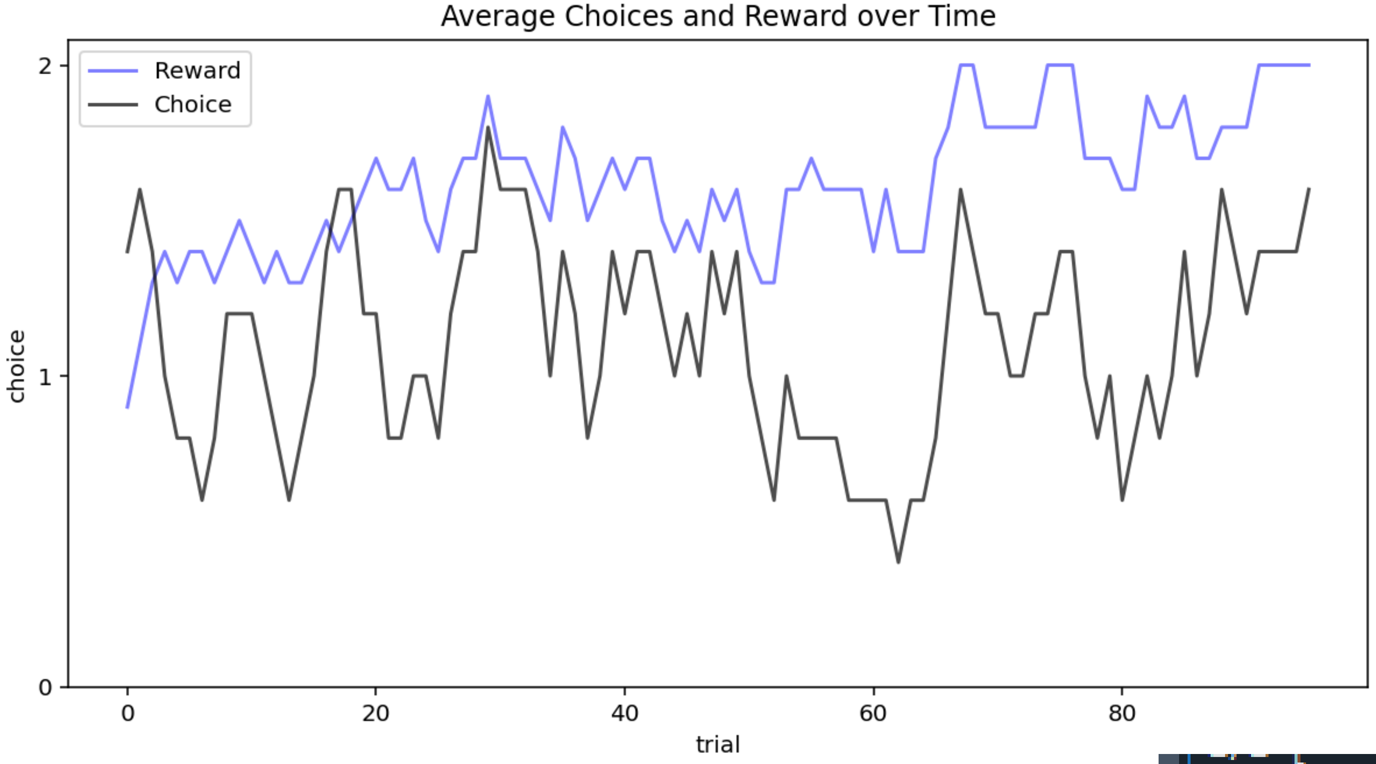
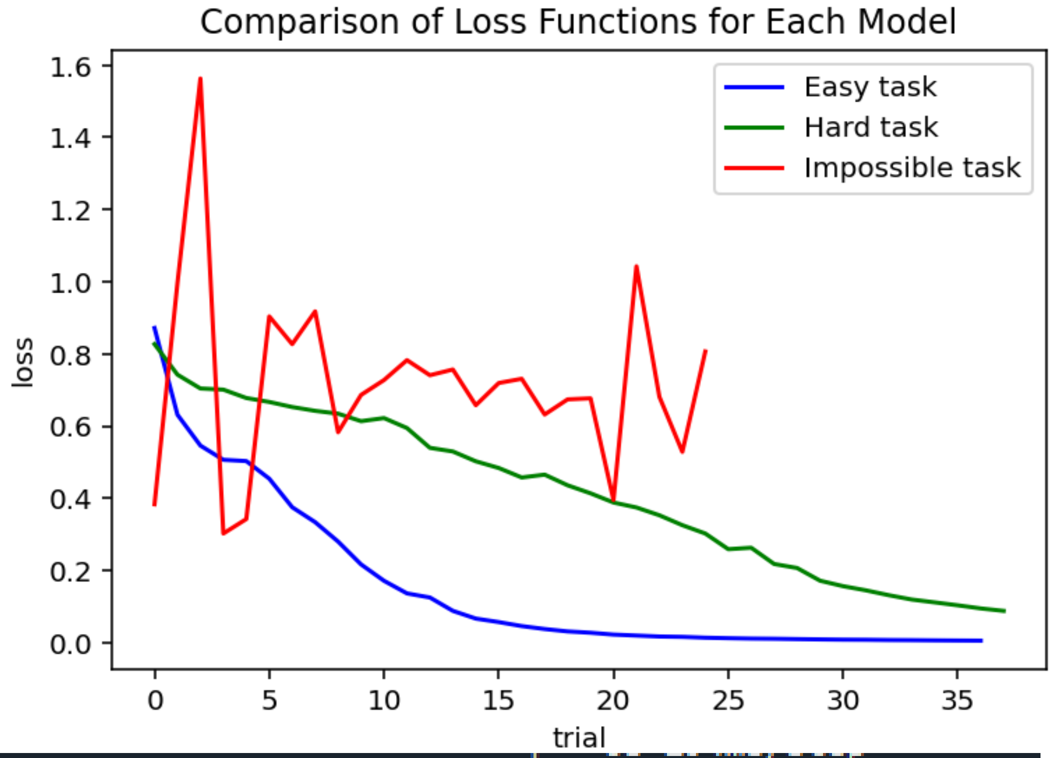
This week’s progress:

- Created a RL task (n-bandit)
- Integrated RL as LVL2 in the main script
- Version 1: maximises accuracy
- Version 2: maximises learning progress

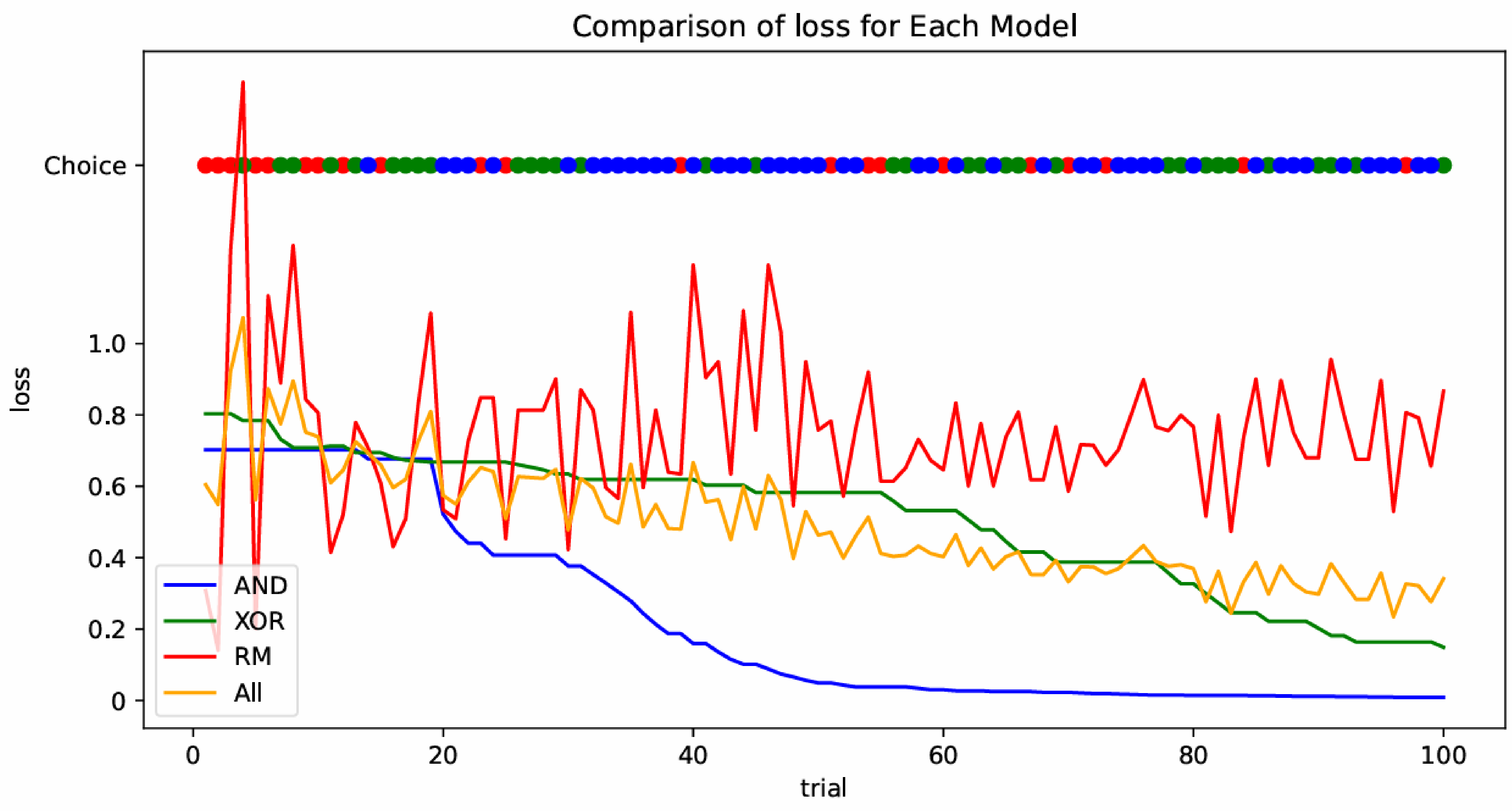
Model maximising accuracy



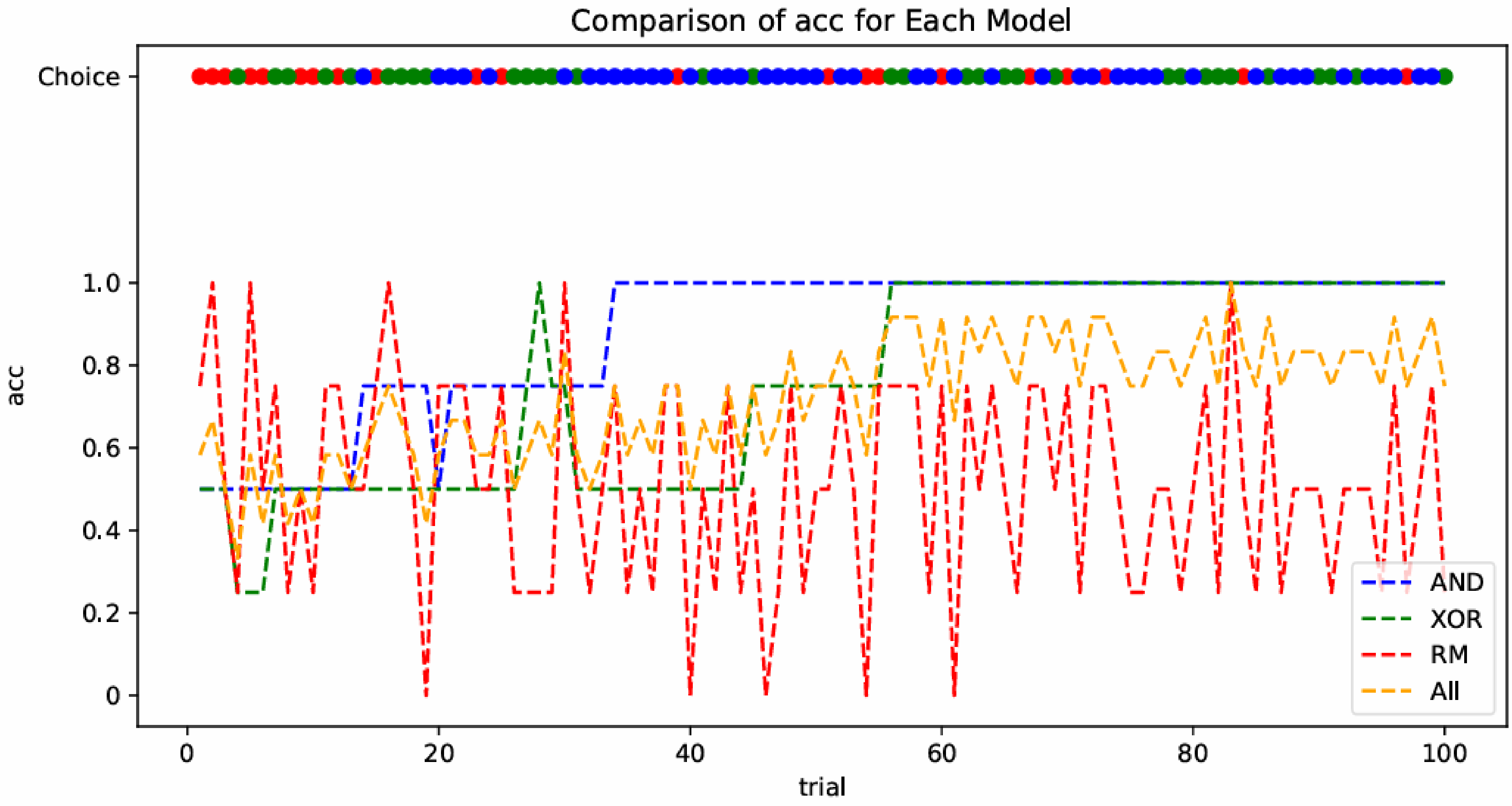



```
['Weights:', array([0.54256715, 0.95444682, 0.97981903])]
Action selection frequency: {0: 25, 1: 38, 2: 37}
Most selected action: 1
mean reward: 0.80
Model name: AND Model, Loss: 0.0059, Accuracy: 1.0000
Model name: XOR Model, Loss: 0.0828, Accuracy: 1.0000
Model name: RM Model, Loss: 0.7818, Accuracy: 0.5000
```

Haopeng's additions

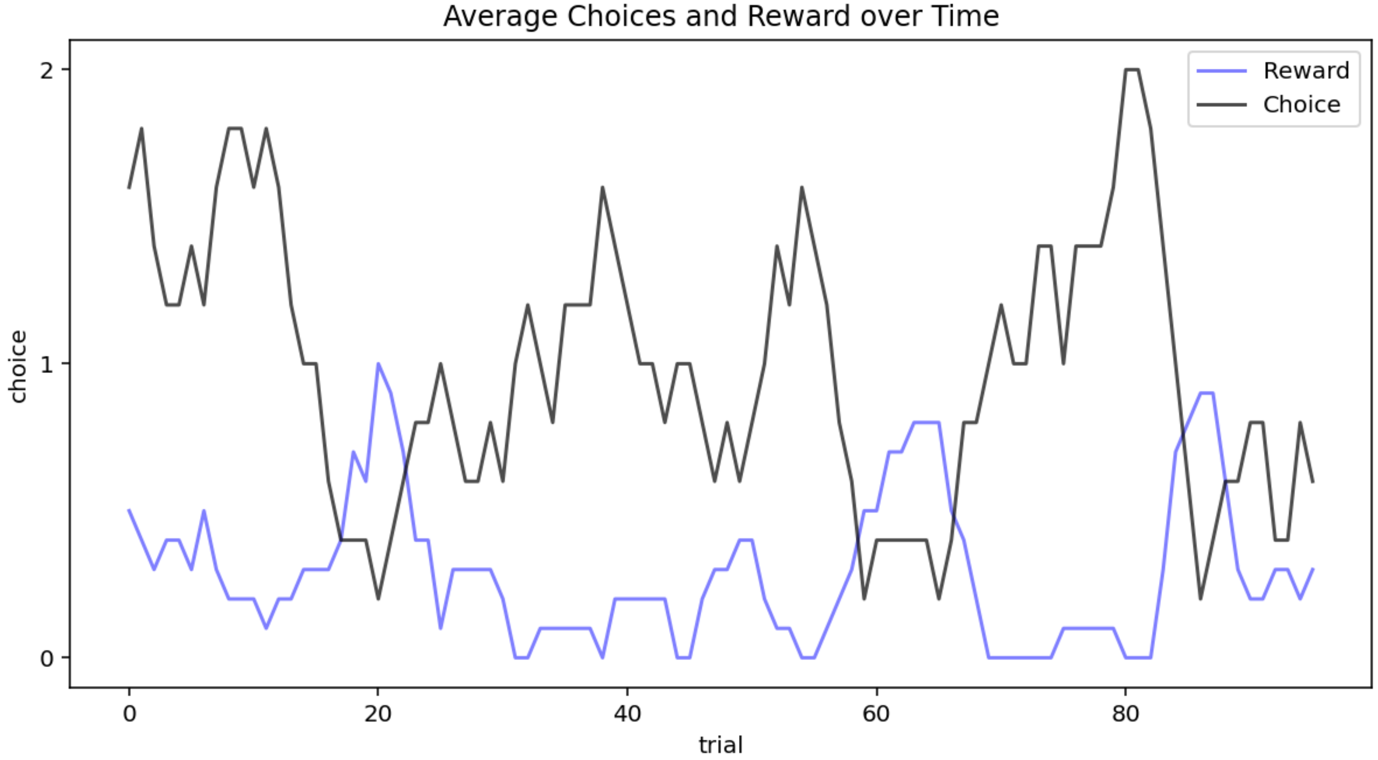


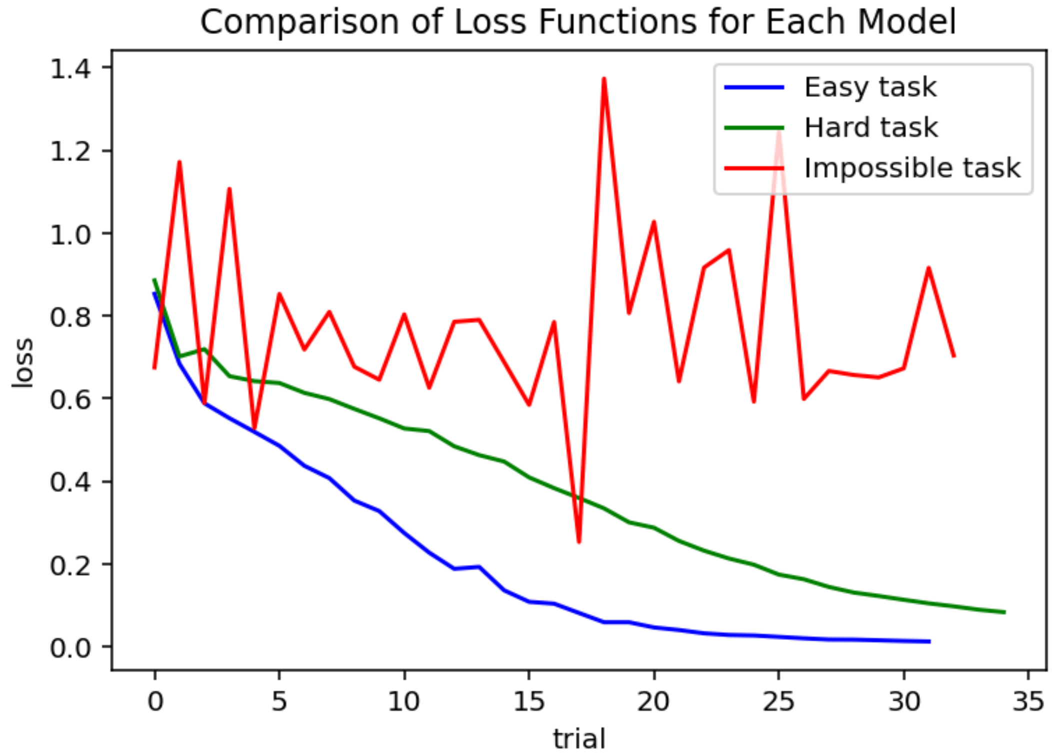
Haopeng's additions



Model maximising learning progress

```
reward = abs(previous_accuracy - trial_accuracy) #the reward is learning progress
reward_list.append(reward)
w[chosen_task] += alpha*(reward - w[chosen_task]) #Rescorla-Wagner learning rule
```






```
['Weights:', array([0.34244747, 0.02661979, 0.02414492])]
Action selection frequency: {0: 33, 1: 35, 2: 32}
Most selected action: 1
mean reward: 0.15
Model name: AND Model, Loss: 0.0117, Accuracy: 1.0000
Model name: XOR Model, Loss: 0.0786, Accuracy: 1.0000
Model name: RM Model, Loss: 0.5649, Accuracy: 0.7500
```

Questions:

- The absolute value of learning progress gives excessive weight to RM. Is it expected, reflects final trials? Fix by focusing on average LP instead?
- How to compare RL models with each other?

Next week’s objectives:

- Smoother plotting and data collection (check Haopeng’s suggestions)
- Integrate all models in level 1 (look into catastrophic interference)
- Compare the effect of tau (reverse temperature)
- Integrate multiple criteria and manually set weights