import traci

from agent import DQNAgent

from simulation import get\_state, CONFIG\_FILE, lanes

def run\_eval(use\_rl=True, steps=200):

agent = DQNAgent(state\_size=5, action\_size=2)

agent.model.load\_weights("trained\_model.h5")

traci.start(["sumo-gui", "-c", CONFIG\_FILE])

total\_waits = []

for step in range(steps):

traci.simulationStep()

state = get\_state()

if use\_rl:

action = agent.act(state)

if action == 1:

current\_phase = traci.trafficlight.getPhase("0")

phases = traci.trafficlight.getCompleteRedYellowGreenDefinition("0")[0].phases

traci.trafficlight.setPhase("0", (current\_phase + 1) % len(phases))

else:

if step % 20 == 0: # naive fixed controller

current\_phase = traci.trafficlight.getPhase("0")

traci.trafficlight.setPhase("0", (current\_phase + 1) % 2)

total\_wait = sum(traci.lane.getLastStepHaltingNumber(l) for l in lanes)

total\_waits.append(total\_wait)

traci.close()

return sum(total\_waits) / steps

if \_\_name\_\_ == "\_\_main\_\_":

print("Avg wait (RL):", run\_eval(use\_rl=True))

print("Avg wait (Fixed):", run\_eval(use\_rl=False))