Deep Learning:
Homework #03

Prof. Tongyao Pang

Deadline: January 4, 2026

Name: **Zhou Shouchen**
Student ID: 2025213446

Problem 1

1: Diffusion Models (5 pts)
Consider a diffusion model defined by the forward SDE

$$\mathrm{d}X_t = -\frac{1}{2}\beta(t)X_t\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}W_t, \qquad t \in [0, T]$$

The corresponding reverse-time SDE, using a learned score function $s_\theta(x,t) \approx \nabla_x \log p_t(x)$, is given by

$$\mathrm{d}X_t = \left[-\frac{1}{2}\beta(t)X_t - \beta(t)s_\theta(X_t, t)\right]\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}\bar{W}_t$$

where $\bar{W}_t$ denotes a standard Brownian motion in reverse time.

- (2 pts) Assume the score function is exact, i.e. $s_\theta(x,t) = \nabla_x \log p_t(x)$. Prove that the solution of the ODE
$$\frac{\mathrm{d}x_t}{\mathrm{d}t} = -\frac{1}{2}\beta(t)x_t - \frac{1}{2}\beta(t)s_\theta(x_t, t)$$
and the reverse-time SDE share the same marginal distribution $p_t(x)$ for all $t \in [0, T]$.
- (3 pts) Use the **integrating factor method** to analytically handle the linear term $-\frac{1}{2}\beta(t)x_t$ in the above ODE, and then apply **forward Euler discretization** to the remaining nonlinear term to derive an explicit one-step update formula for x_{t+h} in terms of x_t.

Solution
(1) We can write the forward SDE in the standard form:

$$\mathrm{d}X_t = f(X_t, t)\mathrm{d}t + g(t)\mathrm{d}W_t, \qquad f(x,t) = -\frac{1}{2}\beta(t)x, \quad g(t) = \sqrt{\beta(t)}$$

Let $p_t(x)$ be the marginal density of the forward process X_t. Then p_t satisfies the forward Fokker-Planck (Kolmogorov forward) equation

$$\partial_t p_t(x) = -\nabla \cdot (f(x,t)p_t(x)) + \frac{1}{2}\nabla \cdot \left(g^2(t)\nabla p_t(x)\right)$$

From the assumption, we know that the score is exact:

$$s_\theta(x,t) = \nabla_x \log p_t(x)$$

Thus the reverse-time SDE written in the usual score-based form is

$$\mathrm{d}X_t = \left[f(X_t, t) - g^2(t)\nabla \log p_t(X_t)\right]\mathrm{d}t + g(t)\mathrm{d}\bar{W}_t$$

By the SDE's property, if the forward SDE

$$\mathrm{d}X_t = f(X_t, t)\mathrm{d}t + g(t)\mathrm{d}W_t$$

has marginal density $p_t(x)$, then the time-reversed dynamics, when initialized from $X_T \sim p_T$ and integrated backward from T to 0, satisfies

$$\mathrm{d}X_t = \left[f(X_t, t) - g^2(t)\nabla_x \log p_t(X_t)\right]\mathrm{d}t + g(t)\mathrm{d}\bar{W}_t$$

where $\bar{W}_t$ is a standard Brownian motion in reverse time. In this case, the one-time marginal distribution of X_t is exactly $p_t(x)$ for all $t \in [0, T]$.
Since the score function is exact, $s_\theta(x,t) = \nabla_x \log p_t(x)$, the reverse-time SDE in the problem has marginals p_t.

Problem 1 continued on next page...

Next we show that the probability flow ODE induces the same marginal density p_t. Combined with the previous step, this implies that the ODE and the reverse-time SDE share the same marginals.

$$\frac{\mathrm{d}x_t}{\mathrm{d}t} = f(x_t, t) - \frac{1}{2}g^2(t)s_\theta(x_t, t) = \underbrace{f(x, t) - \frac{1}{2}g^2(t)\nabla_x \log p_t(x)}_{v(x,t)}$$

And let $\rho_t(x)$ be the density induced by the ODE flow.
For deterministic dynamics $\dot{x} = v(x, t)$, the density ρ_t satisfies the continuity (Liouville) equation

$$\partial_t \rho_t(x) = -\nabla \cdot (v(x, t)\rho_t(x))$$

On the other hand, as shown above, the forward diffusion marginals p_t satisfy

$$\begin{aligned}\partial_t p_t(x) &= -\nabla \cdot (f(x, t)p_t(x)) + \frac{1}{2}\nabla \cdot \left(g^2(t)\nabla p_t(x)\right)\\ &= -\nabla \cdot (f(x, t)p_t(x)) + \frac{1}{2}\nabla \cdot \left(g^2(t)\nabla_x \log p_t(x) \cdot p_t(x)\right)\\ &= -\nabla \cdot \left(\left[f(x, t) - \frac{1}{2}g^2(t)\nabla_x \log p_t(x)\right] p_t(x)\right)\\ &= -\nabla \cdot (v(x, t)p_t(x))\end{aligned}$$

Therefore, both ρ_t and p_t satisfy the same continuity equation

$$\partial_t \rho_t(x) = -\nabla \cdot (v(x, t)\rho_t(x)) \,.$$

If $\rho_{t_0} = p_{t_0}$ at some time t_0, then under standard regularity assumptions on $v(x, t)$, uniqueness of solutions to the continuity equation implies

$$\rho_t(x) = p_t(x) \qquad \text{for all } t$$

In sampling, one takes $t_0 = T$, draws $x_T \sim p_T$, and integrates the ODE from T to 0 using a negative step size.
In practice, sampling from p_T to p_0 is achieved by integrating this same ODE backward in time using a negative step size, which does not require posing a terminal-value problem for the PDE.
So above all, the reverse-time SDE has marginal density p_t at each time t (when run from T to 0). Moreover, the probability flow ODE induces a density ρ_t that satisfies the same continuity equation as p_t.
Since $\rho_{t_0} = p_{t_0}$ at some time t_0 ($t_0 = T$ in sampling), uniqueness of solutions to the continuity equation with initial time t_0 implies that $\rho_t = p_t$ for all $t \in [0, T]$.
(2) For the probability flow ODE with exact score $s_\theta(x, t) = \nabla_x \log p_t(x)$:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = -\frac{1}{2}\beta(t)x - \frac{1}{2}\beta(t)s_\theta(x, t)$$

Rewrite it as a linear ODE plus a nonlinear forcing term:

$$\frac{\mathrm{d}x}{\mathrm{d}t} + \underbrace{\frac{1}{2}\beta(t)}_{P(t)} x(t) = -\frac{1}{2}\beta(t)s_\theta(x(t), t)$$

Apply the integrating factor method, the integrating factor is

$$\mu(t) = \exp\left(\int_0^t P(u)\mathrm{d}u\right) = \exp\left(\int_0^t \frac{1}{2}\beta(u)\mathrm{d}u\right)$$

Then

$$\begin{aligned}
\mu(t)\frac{\mathrm{d}x}{\mathrm{d}t}+\mu(t)P(t)x(t) &= \mu(t)\cdot\left(-\frac{1}{2}\beta(t)s_\theta(x(t),t)\right)\\
\Rightarrow\quad \frac{\mathrm{d}}{\mathrm{d}t}\left(\mu(t)x(t)\right) &= -\frac{1}{2}\mu(t)\beta(t)s_\theta(x(t),t)
\end{aligned}$$

Let $y(t)=\mu(t)x(t), y'(t)=-\frac{1}{2}\mu(t)\beta(t)s_\theta(x(t),t)$. Then

$$\frac{\mathrm{d}y}{\mathrm{d}t}=-\frac{1}{2}\mu(t)\beta(t)s_\theta(x(t),t)$$

Then we can apply the forward Euler method to the remaining term:

$$y(t+h)\approx y(t)+hy'(t)=y(t)-\frac{1}{2}h\mu(t)\beta(t)s_\theta(x(t),t)$$

Substitute back $y(t)=\mu(t)x(t)$ and divide by $\mu(t+h)$:

$$\begin{aligned}
\mu(t+h)x(t+h) &\approx \mu(t)x(t)-\frac{1}{2}h\mu(t)\beta(t)s_\theta(x(t),t)\\
\Rightarrow\quad x(t+h) &\approx \frac{\mu(t)}{\mu(t+h)}x(t)-\frac{1}{2}h\frac{\mu(t)}{\mu(t+h)}\beta(t)s_\theta(x(t),t)
\end{aligned}$$

Finally, using the definition of $\mu(t)$:

$$\begin{aligned}
\frac{\mu(t)}{\mu(t+h)} &= \frac{\exp\left(\int_0^t P(u)\mathrm{d}u\right)}{\exp\left(\int_0^{t+h} P(u)\mathrm{d}u\right)}\\
&= \exp\left(-\int_t^{t+h} P(u)\mathrm{d}u\right)\\
&= \exp\left(-\int_t^{t+h} \frac{1}{2}\beta(u)\mathrm{d}u\right)\\
&= \exp\left(-\frac{1}{2}\int_t^{t+h} \beta(u)\mathrm{d}u\right)\\
&\approx \exp\left(-\frac{1}{2}\beta(t)\cdot h\right)
\end{aligned}$$

So above all, the explicit one-step update formula is

$$x_{t+h}\approx\exp\left(-\frac{1}{2}\beta(t)\cdot h\right)x_t-\frac{1}{2}h\beta(t)\exp\left(-\frac{1}{2}\beta(t)\cdot h\right)s_\theta(x_t,t)$$

Problem 2

2: Policy Improvement in Value Iteration and Policy Iteration (3 pts)

- (2 pts) Prove that the policy improvement step in in **Policy iteration** guarantees a monotonic improvement in the value function, i.e.,

$$V^{\pi_{k+1}}(s) \geq V^{\pi_k}(s), \quad \forall s \in \mathcal{S}$$

- (3 pts) Let π_k be the greedy policy extracted from the k-th iteration of the **value iteration** algorithm, i.e.,

$$\pi_k(s) = \operatorname*{argmax}_a \sum_{s'} P(s', r|s, a) \left[r(s, a, s') + \gamma V_k(s')\right]$$

  Let π_{k+1} be the policy extracted from the $(k+1)$-th value iteration step. Is it still true that the value function improves under the new policy, i.e.,

$$V^{\pi_{k+1}}(s) \geq V^{\pi_k}(s), \quad \forall s \in \mathcal{S}?$$

  Either **prove** the statement or provide a **counterexample** to disprove it.

Solution

For a discounted finite MDP with state space $\mathcal{S}$, action space $\mathcal{A}$, transition dynamics $P(s', r|s, a)$, and discount factor $\gamma \in (0, 1)$. For a policy π, the value and action-value functions are

$$V^\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} \middle| S_0 = s \right], \qquad Q^\pi(s, a) = \sum_{s', r} P(s', r|s, a) \left[r + \gamma V^\pi(s')\right]$$

And the Bellman equation is that:

$$V^\pi = \gamma P^\pi V^\pi + b^\pi$$

where for each state s,

$$b^\pi(s) = \sum_a \pi(a|s) \sum_{s', r} P(s', r|s, a) r, \qquad [P^\pi]_{ss'} = \sum_a \pi(a|s) \sum_r P(s', r|s, a)$$

(1) For policy iteration, at iteration k, it follows:
1. Policy evaluation: compute V^{π_k} exactly.
2. Policy improvement: $\pi_{k+1}(s) \in \operatorname*{argmax}_a Q^{\pi_k}(s, a)$.

From the policy improvement, we can get that for every s,

$$\sum_a \pi_{k+1}(a|s) Q^{\pi_k}(s, a) \geq \sum_a \pi_k(a|s) Q^{\pi_k}(s, a) = V^{\pi_k}(s)$$

Using the definition of Q^{π_k}, the left-hand side equals

$$\sum_a \pi_{k+1}(a|s) \sum_{s', r} P(s', r|s, a) \left[r + \gamma V^{\pi_k}(s')\right] = b^{\pi_{k+1}}(s) + \gamma \sum_{s'} [P^{\pi_{k+1}}]_{ss'} V^{\pi_k}(s')$$

Write it in the vector form:

$$b^{\pi_{k+1}} + \gamma P^{\pi_{k+1}} V^{\pi_k} \geq V^{\pi_k} \quad \Rightarrow \quad (I - \gamma P^{\pi_{k+1}}) V^{\pi_k} \leq b^{\pi_{k+1}}$$

On the other hand, by the Bellman equation for π_{k+1} is

$$(I - \gamma P^{\pi_{k+1}}) V^{\pi_{k+1}} = b^{\pi_{k+1}} \quad \Rightarrow \quad V^{\pi_{k+1}} = (I - \gamma P^{\pi_{k+1}})^{-1} b^{\pi_{k+1}}$$

Problem 2 continued on next page...

Since $\gamma \in (0,1)$ and $P^{\pi_{k+1}}$ is a stochastic matrix, we have

$$(I-\gamma P^{\pi_{k+1}})^{-1} = \sum_{t=0}^{\infty} (\gamma P^{\pi_{k+1}})^t$$

whose entries are all nonnegative. Thus the vector form of policy improvement step could be further written as

$$V^{\pi_k} \leq (I-\gamma P^{\pi_{k+1}})^{-1} b^{\pi_{k+1}} = V^{\pi_{k+1}}$$

So above all, we have proved that

$$V^{\pi_{k+1}}(s) \geq V^{\pi_k}(s), \qquad \forall s \in \mathcal{S}$$

(2) For value iteration, we can construct a following counterexample:
Let $\gamma = 0.9$ and $\mathcal{S} = \{s_0, s_1\}, \mathcal{A} = a_0, a_1$. The transition and reward dynamics are deterministic and given by the joint distribution $P(s', r|s, a)$:
At state s_0:

$$P(s_0, 1|s_0, a_0) = 1, \qquad P(s_1, 2|s_0, a_1) = 1$$

At state s_1 (only one valid action a_0):

$$P(s_0, 0|s_1, a_0) = 1$$

All other transition probabilities are zero.
Thus the value iteration updates the value function according to

$$V_{k+1}(s) = \max_a \sum_{s',r} P(s', r|s, a)\left[r + \gamma V_k(s')\right]$$

and the greedy policy

$$\pi_k(s) \in \operatorname*{argmax}_a \sum_{s',r} P(s', r|s, a)\left[r + \gamma V_k(s')\right]$$

Initialize $V_0(s_0) = V_0(s_1) = 0$.
In this construction, at state s_0, $r + \gamma V_0(s')$ has

$$1 + \gamma V_0(s_0) = 1, \qquad 2 + \gamma V_0(s_1) = 2$$

so $\pi_0(s_0) = a_1$. At state s_1, the only action is a_0, hence $\pi_0(s_1) = a_0$.
For the first interation's update, compute V_1:

$$\begin{aligned} V_1(s_0) &= \max\{1 + \gamma V_0(s_0), 2 + \gamma V_0(s_1)\} = 2 \\ V_1(s_1) &= 0 + \gamma V_0(s_0) = 0 \end{aligned}$$

And the corresponding greedy policy is:

$$1 + \gamma V_1(s_0) = 1 + 0.9 \cdot 2 = 2.8, \qquad 2 + \gamma V_1(s_1) = 2$$

Thus $\pi_1(s_0) = a_0$, and $\pi_1(s_1) = a_0$.
But for the true value function: with policy π_1, from s_0 the agent stays in s_0 and receives reward 1 at every step:

$$\begin{aligned} V^{\pi_1}(s_0) &= \sum_{t=0}^{\infty} \gamma^t \cdot 1 = \frac{1}{1-\gamma} = \frac{1}{0.1} = 10 \\ V^{\pi_1}(s_1) &= 0 + \gamma V^{\pi_1}(s_0) = 0.9 \cdot 10 = 9 \end{aligned}$$

With policy π_0, starting from s_0 the reward sequence is $2, 0, 2, 0, \ldots$. Thus

$$V^{\pi_0}(s_0) = 2 + \gamma^2 \cdot 2 + \gamma^4 \cdot 2 + \cdots = 2\sum_{t=0}^{\infty} \gamma^{2t} = \frac{2}{1-\gamma^2} = \frac{2}{1-0.81} = \frac{2}{0.19} \approx 10.526$$
$$V^{\pi_0}(s_1) = 0 + \gamma V^{\pi_0}(s_0) = 0.9 \cdot \frac{2}{0.19} \approx 9.474$$

Since

$$V^{\pi_1}(s_0) = 10 < 10.526 \approx V^{\pi_0}(s_0), \qquad V^{\pi_1}(s_1) = 9 < 9.474 \approx V^{\pi_0}(s_1)$$

Thus, in this construction,

$$V^{\pi_1}(s_0) < V^{\pi_0}(s_0) \quad \text{and} \quad V^{\pi_1}(s_1) < V^{\pi_0}(s_1)$$

Which means that it is not true that

$$V^{\pi_{k+1}}(s) \geq V^{\pi_k}(s), \quad \forall s \in \mathcal{S}$$

Thus we have a counterexample to disprove the statement.

Problem 3

3: Policy Gradient Method (7 pts)
Consider a discounted Markov Decision Process (MDP) $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$ with states $s \in \mathcal{S}$, actions $a \in \mathcal{A}$, transition kernel $P(s'|s,a)$, reward $r(s,a)$ bounded, and discount $\gamma \in (0,1)$. Let $\pi_\theta(a|s)$ be a differentiable, stochastic policy with parameters θ, and let s_0 be a fixed start state. Define the (discounted) return

$$G_0 = \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)$$

and the performance objective

$$J(\theta) = V^{\pi_\theta}(s_0) = \mathbb{E}_{\tau \sim \pi_\theta}[G_0]$$

where a trajectory $\tau = (s_0, a_0, s_1, a_1, \ldots)$ is generated by $s_{t+1} \sim P(\cdot|s_t, a_t)$ and $a_t \sim \pi_\theta(\cdot|s_t)$.
Denote the value and action-value functions by

$$V^\pi(s) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \Big| s_0 = s\right], \qquad Q^\pi(s,a) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \Big| s_0 = s, a_0 = a\right]$$

and the advantage by $A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s)$. Let $d^\pi(s)$ be the (unnormalized) γ-discounted state visitation distribution:

$$d^\pi(s) = \sum_{t=0}^{\infty} \gamma^t \Pr(s_t = s|\pi)$$

- (4 pts) Prove the Policy Gradient Theorem.

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t \nabla_\theta \log \pi_\theta(a_t|s_t) Q^{\pi_\theta}(s_t, a_t)\right] = \frac{1}{1-\gamma} \mathbb{E}_{s \sim d^{\pi_\theta}, a \sim \pi_\theta}\left[\nabla_\theta \log \pi_\theta(a|s) Q^{\pi_\theta}(s,a)\right]$$

- (3 pts) Show that for any function $b : \mathcal{S} \to \mathbb{R}$,

$$\mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t \nabla_\theta \log \pi_\theta(a_t|s_t) b(s_t)\right] = 0$$

and hence the policy gradient can be equivalently written as

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t \nabla_\theta \log \pi_\theta(a_t|s_t) \left(Q^{\pi_\theta}(s_t, a_t) - b(s_t)\right)\right]$$

In the advantage method, how should $b(s)$ be chosen, and what is the benefit of this choice?

Solution
(1) Let $\tau = (s_0, a_0, s_1, a_1, \ldots)$ be an trajectory with initial state s_0. Since it is a Markov model, thus its probability under π_θ is

$$p_\theta(\tau) = \prod_{t=0}^{\infty} \pi_\theta(a_t|s_t) P(s_{t+1}|s_t, a_t)$$

Since the transition probability is independent of θ, thus we have

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[G_0] = \int G_0(\tau) p_\theta(\tau) \mathrm{d}\tau$$

$$\Rightarrow \quad \nabla_\theta J(\theta) = \int G_0(\tau) \nabla_\theta p_\theta(\tau) \mathrm{d}\tau = \int G_0(\tau) p_\theta(\tau) \nabla_\theta \log p_\theta(\tau) \mathrm{d}\tau = \mathbb{E}_{\tau \sim \pi_\theta}\left[\nabla_\theta \log p_\theta(\tau) G_0\right]$$

Problem 3 continued on next page...

Using the definition of the discounted return, for each t, we have

$$\begin{aligned}
G_0 &= \sum_{k=0}^{\infty} \gamma^k r(s_k, a_k) \\
&= \sum_{k=0}^{t-1} \gamma^k r(s_k, a_k) + \sum_{k=t}^{\infty} \gamma^t \gamma^{k-t} r(s_k, a_k) \\
&= \sum_{k=0}^{t-1} \gamma^k r(s_k, a_k) + \gamma^t \sum_{k=0}^{\infty} \gamma^k r(s_{k+t}, a_{k+t}) \\
&= \underbrace{\sum_{k=0}^{t-1} \gamma^k r(s_k, a_k)}_{=R_{<t}} + \gamma^t G_t
\end{aligned}$$

Then

$$\mathbb{E}_{\tau\sim\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a_t|s_t) G_0\right] = \mathbb{E}_{\tau\sim\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a_t|s_t) R_{<t}\right] + \mathbb{E}_{\tau\sim\pi_\theta}\left[\gamma^t \nabla_\theta \log \pi_\theta(a_t|s_t) G_t\right]$$

Let the history up to state s_t be $H_t = (s_0, a_0, \ldots, s_t)$. Note that $R_{<t}$ is measurable with respect to H_t and does not depend on a_t. Using the law of iterative expectation, we can get that

$$\mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a_t|s_t) R_{<t}\right] = \mathbb{E}_{\pi_\theta}\left[\mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a_t|s_t) R_{<t} | H_t\right]\right] = \mathbb{E}_{\pi_\theta}\left[R_{<t} \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a_t|s_t) | H_t\right]\right]$$

Conditioned on H_t, the state s_t is fixed and $a_t \sim \pi_\theta(\cdot|s_t)$, thus

$$\mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a_t|s_t) | H_t\right] = \sum_a \pi_\theta(a|s_t) \nabla_\theta \log \pi_\theta(a|s_t) = \sum_a \nabla_\theta \pi_\theta(a|s_t) = \nabla_\theta \sum_a \pi_\theta(a|s_t) = \nabla_\theta 1 = 0$$

i.e. $\mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(a_t|s_t) R_{<t}] = 0$, so we can get that

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t \nabla_\theta \log \pi_\theta(a_t|s_t) G_t\right]$$

From the definition of Q^{π_θ}, we have

$$Q^{\pi_\theta}(s_t, a_t) = \mathbb{E}_{\pi_\theta}[G_t | s_t, a_t]$$

Using the law of total expectation to the above result again, we can get that

$$\begin{aligned}
\mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a_t|s_t) G_t\right] &= \mathbb{E}_{\pi_\theta}\left[\mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(a_t|s_t) G_t | s_t, a_t]\right] \\
&= \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a_t|s_t) \mathbb{E}_{\pi_\theta}[G_t | s_t, a_t]\right] \\
&= \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a_t|s_t) Q^{\pi_\theta}(s_t, a_t)\right] \\
\Rightarrow \quad \nabla_\theta J(\theta) &= \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t \nabla_\theta \log \pi_\theta(a_t|s_t) G_t\right] \\
&= \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a_t|s_t) G_t\right] \\
&= \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a_t|s_t) Q^{\pi_\theta}(s_t, a_t)\right] \\
&= \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t \nabla_\theta \log \pi_\theta(a_t|s_t) Q^{\pi_\theta}(s_t, a_t)\right]
\end{aligned}$$

Which means that we have proved the first part of the equality.

Problem 3 continued on next page...

For the second part, expand the expectation, we can get that for any function $f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$:

$$\begin{aligned}
\mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty}\gamma^t f(s_t,a_t)\right] &= \sum_{t=0}^{\infty}\gamma^t \sum_{s}\Pr(s_t=s|\pi_\theta)\sum_{a}\pi_\theta(a|s)f(s,a)\\
&= \sum_{s}\sum_{t=0}^{\infty}\gamma^t \Pr(s_t=s|\pi_\theta)\sum_{a}\pi_\theta(a|s)f(s,a)\\
&= \sum_{s}d^{\pi_\theta}(s)\sum_{a}\pi_\theta(a|s)f(s,a)
\end{aligned}$$

Put $f(s,a)=\nabla_\theta \log \pi_\theta(a|s)Q^{\pi_\theta}(s,a)$ into above equation, we can get that

$$\nabla_\theta J(\theta) = \sum_{s}d^{\pi_\theta}(s)\sum_{a}\pi_\theta(a|s)\nabla_\theta \log\pi_\theta(a|s)Q^{\pi_\theta}(s,a)$$

Also, the normalization factor of sampling s from d^{π_θ} is

$$\sum_{s}d^{\pi_\theta}(s) = \sum_{t=0}^{\infty}\gamma^t\sum_{s}\Pr(s_t=s|\pi_\theta) = \sum_{t=0}^{\infty}\gamma^t = \frac{1}{1-\gamma}$$

Which means that the probability of sampling s is

$$\Pr_{s\sim d^{\pi_\theta}}(s) = (1-\gamma)d^{\pi_\theta}(s)$$

Therefore the previous identity can be rewritten as

$$\begin{aligned}
\nabla_\theta J(\theta) &= \sum_{s}d^{\pi_\theta}(s)\sum_{a}\pi_\theta(a|s)\nabla_\theta\log\pi_\theta(a|s)Q^{\pi_\theta}(s,a)\\
&= \frac{1}{1-\gamma}\sum_{s}(1-\gamma)d^{\pi_\theta}(s)\sum_{a}\pi_\theta(a|s)\nabla_\theta\log\pi_\theta(a|s)Q^{\pi_\theta}(s,a)\\
&= \frac{1}{1-\gamma}\sum_{s}\Pr_{s\sim d^{\pi_\theta}}(s)\sum_{a}\pi_\theta(a|s)\left[\nabla_\theta\log\pi_\theta(a|s)Q^{\pi_\theta}(s,a)\right]\\
&= \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi_\theta},a\sim\pi_\theta}\left[\nabla_\theta\log\pi_\theta(a|s)Q^{\pi_\theta}(s,a)\right]
\end{aligned}$$

So above all, we have proved that

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty}\gamma^t\nabla_\theta\log\pi_\theta(a_t|s_t)Q^{\pi_\theta}(s_t,a_t)\right] = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi_\theta},a\sim\pi_\theta}\left[\nabla_\theta\log\pi_\theta(a|s)Q^{\pi_\theta}(s,a)\right]$$

(2) Using the law of iterative expectation, we have

$$\mathbb{E}_{\pi_\theta}\left[\nabla_\theta\log\pi_\theta(a_t|s_t)b(s_t)\right] = \mathbb{E}_{\pi_\theta}\left[\mathbb{E}_{\pi_\theta}\left[\nabla_\theta\log\pi_\theta(a_t|s_t)b(s_t)|s_t\right]\right] = \mathbb{E}_{\pi_\theta}\left[b(s_t)\mathbb{E}_{\pi_\theta}\left[\nabla_\theta\log\pi_\theta(a_t|s_t)|s_t\right]\right]$$

For any state s,

$$\mathbb{E}_{\pi_\theta}\left[\nabla_\theta\log\pi_\theta(a_t|s_t)|s_t\right] = \sum_{a}\pi_\theta(a|s_t)\nabla_\theta\log\pi_\theta(a|s_t) = \sum_{a}\nabla_\theta\pi_\theta(a|s_t) = \nabla_\theta\sum_{a}\pi_\theta(a|s_t) = \nabla_\theta 1 = 0$$

i.e.

$$\mathbb{E}_{\pi_\theta}\left[\nabla_\theta\log\pi_\theta(a_t|s_t)b(s_t)\right] = \mathbb{E}_{\pi_\theta}\left[b(s_t)\mathbb{E}_{\pi_\theta}\left[\nabla_\theta\log\pi_\theta(a_t|s_t)|s_t\right]\right] = \mathbb{E}_{\pi_\theta}\left[b(s_t)\cdot 0\right] = 0$$

Thus each term is zero, so summing up, we can get that

$$\mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty}\gamma^t\nabla_\theta\log\pi_\theta(a_t|s_t)b(s_t)\right] = \sum_{t=0}^{\infty}\gamma^t\mathbb{E}_{\pi_\theta}\left[\nabla_\theta\log\pi_\theta(a_t|s_t)b(s_t)\right] = \sum_{t=0}^{\infty}\gamma^t\cdot 0 = 0$$

Problem 3 continued on next page...

Subtracting this zero term from the policy gradient, we can get that

$$\begin{aligned}
\nabla_\theta J(\theta) &= \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty}\gamma^t\nabla_\theta\log\pi_\theta(a_t|s_t)Q^{\pi_\theta}(s_t,a_t)\right] \\
&= \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty}\gamma^t\nabla_\theta\log\pi_\theta(a_t|s_t)Q^{\pi_\theta}(s_t,a_t)\right] - \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty}\gamma^t\nabla_\theta\log\pi_\theta(a_t|s_t)b(s_t)\right] \\
&= \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty}\gamma^t\nabla_\theta\log\pi_\theta(a_t|s_t)\left(Q^{\pi_\theta}(s_t,a_t)-b(s_t)\right)\right]
\end{aligned}$$

So above all, we have proved that

$$\mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty}\gamma^t\nabla_\theta\log\pi_\theta(a_t|s_t)b(s_t)\right] = 0$$

and

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty}\gamma^t\nabla_\theta\log\pi_\theta(a_t|s_t)\left(Q^{\pi_\theta}(s_t,a_t)-b(s_t)\right)\right]$$

In the advantage method, choose

$$b(s) = V^{\pi_\theta}(s)$$

so that $Q^{\pi_\theta}(s,a) - b(s) = Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s) = A^{\pi_\theta}(s,a)$ and

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty}\gamma^t\nabla_\theta\log\pi_\theta(a_t|s_t)A^{\pi_\theta}(s_t,a_t)\right]$$

This choice keeps the estimator unbiased and most importantly reducing variance, which makes the training more stable and sample-efficient policy optimization.

Problem 4

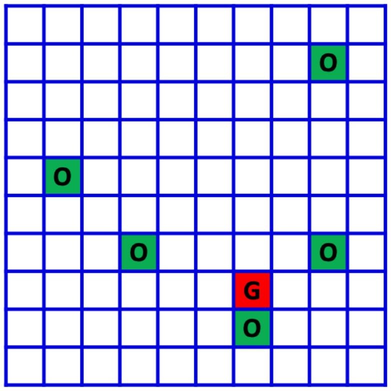
Programming Problem: Comparing SARSA and Q-Learning on 10 × 10 Gridworld (10 pts)
In this problem, you will implement and compare two temporal-difference reinforcement learning algorithms:

- SARSA (On-policy TD(0))
- Q-Learning (Off-policy TD(0))

You will test them on a 10 × 10 Gridworld with obstacles and a goal.

Environment Description

- **Grid size**: 10×10 (states indexed by (row, col))
- **Actions**: {up, down, left, right}
- **Transition**: deterministic, constrained by grid boundaries
- **Rewards**:
    - Reaching the **goal state** (red G): **+10**, then episode ends
    - Stepping into an **obstacle** (green O): **-10**, and the agent remains in the obstacle
    - All other steps: **0**
- **Goal state behavior**:
    - It is **terminal and absorbing**: after entering the goal, any action returns the agent to the goal state with **reward 0**
- **Start state**: always starts at $(0, 0)$
- **Discount factor**: $\gamma = 0.99$



Task: Compare Convergence

1. Implement **SARSA** and **Q-learning** using an ϵ-greedy policy with $\epsilon = 0.1$ or 0.5.
2. Randomly select **10 distinct non-terminal, non-obstacle states** from the grid.
3. During training, track the **value estimate** at each selected state:
    - Define the state value as $V(s) = \max_a Q(s, a)$.

Problem 4 continued on next page...

4. For each algorithm, plot the **sum of the values** at the 10 selected states over episodes and compare the convergence rate.

Solution

For the details of the implementation, set the left-up corner to be $(0,0)$, right-down corner to be $(9,9)$.
And when implement the algorithms, set the learning rate to be $\alpha \leftarrow 0.2$, the total number of episodes to be 3000. And for Q-Learning, the max steps is set to be 500.
For reproducibility, set the same seed 0. Then the sum of the values at the 10 selected states over episodes could seen in Figure 1. We can find that for both $\epsilon = 0.1$ and $\epsilon = 0.5$, Q-learning converges faster and reaches higher summed state values than SARSA. The gap is more pronounced for larger ϵ, as SARSA's on-policy updates account for exploratory actions and slow convergence.

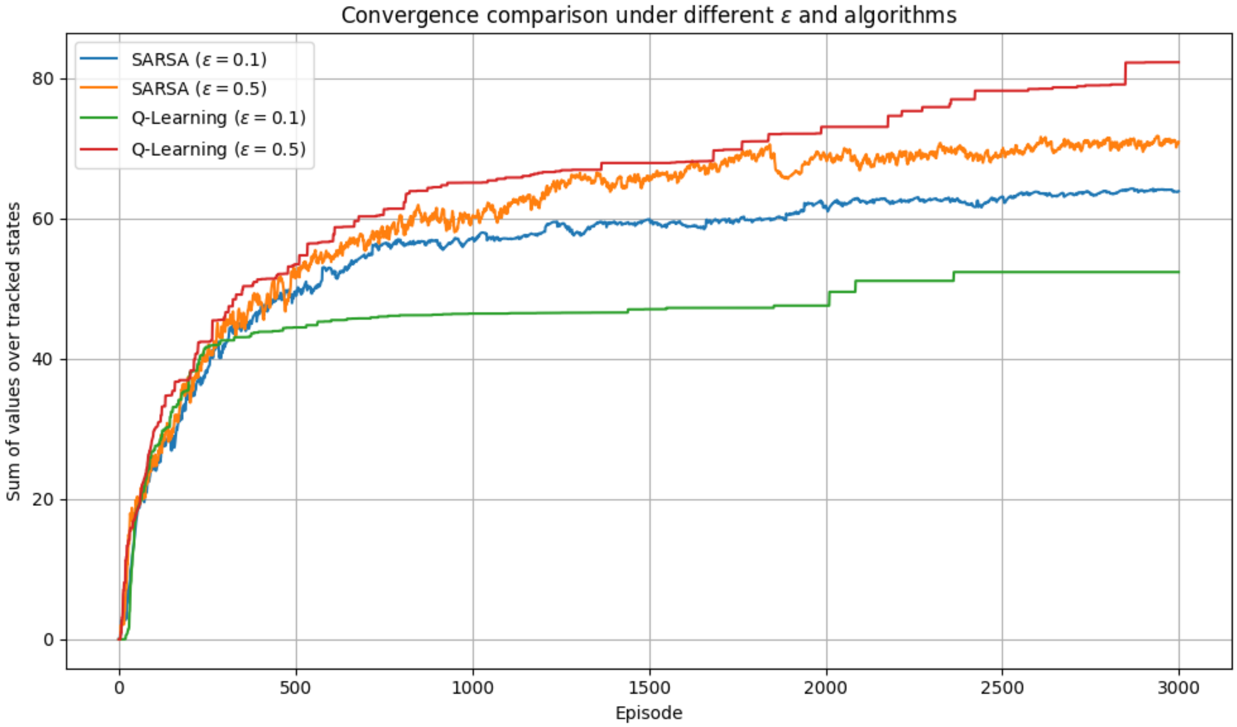


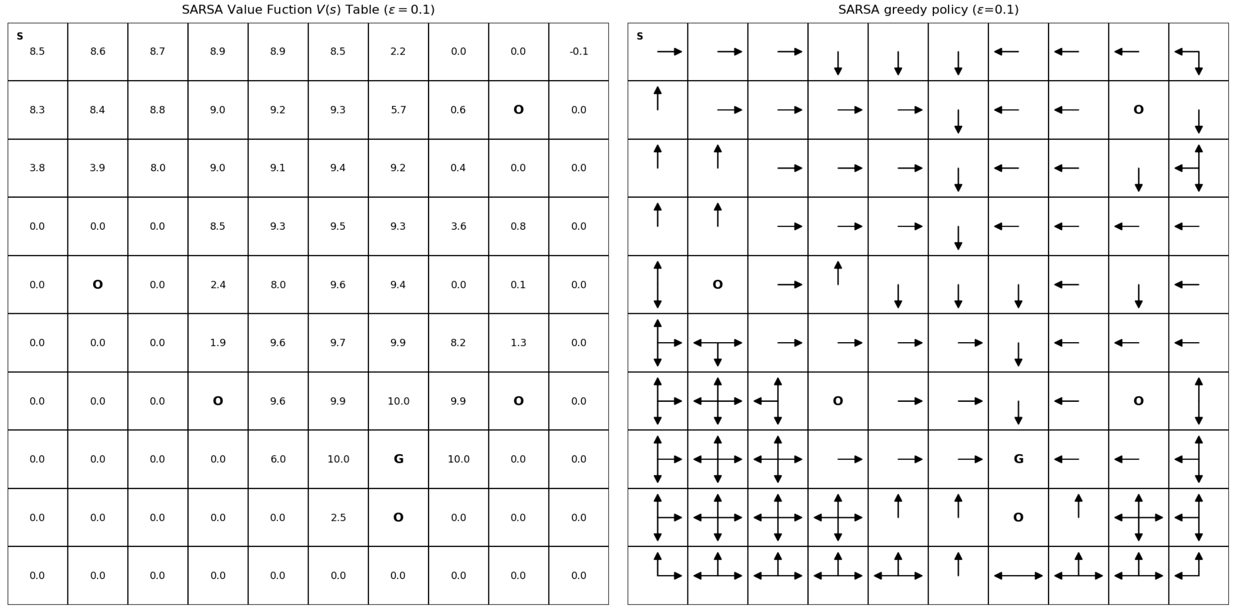
Figure 1: Convergence of SARSA and Q-learning on the 10×10 Gridworld under different exploration rates.

The value function $V(s)$ and the policy π of different algorithms and ϵ are shown in Figure 2, Figure 3, Figure 4, and Figure 5.

From the results, we could find that under a higher exploration rate ($\epsilon = 0.5$), Q-learning explores the grid more thoroughly and therefore assigns non-zero values to many more states; in contrast, with lower exploration or more conservative learning, a noticeable portion of states can remain effectively unvisited, resulting in $V(s) \approx 0$ across those regions. Empirically, the summed values over the tracked states increase faster and converge to a higher level for Q-learning than for SARSA, especially when ϵ is large.

And Q-learning is off-policy that updates toward the greedy target $r + \gamma \max_{a'} Q(s', a')$, so even when behavior is highly exploratory ($\epsilon = 0.5$), the value estimates are still driven by the optimal continuation.

SARSA is on-policy and updates using the next executed action $r + \gamma Q(s', a')$ where a' is sampled from the ϵ-greedy behavior; with $\epsilon = 0.5$, the probability of taking suboptimal actions is high, so SARSA learns a more conservative policy and lower state values. As a result, Q-learning retains faster convergence and higher final values under heavy exploration, while SARSA's estimates are more strongly degraded by the exploratory behavior itself.



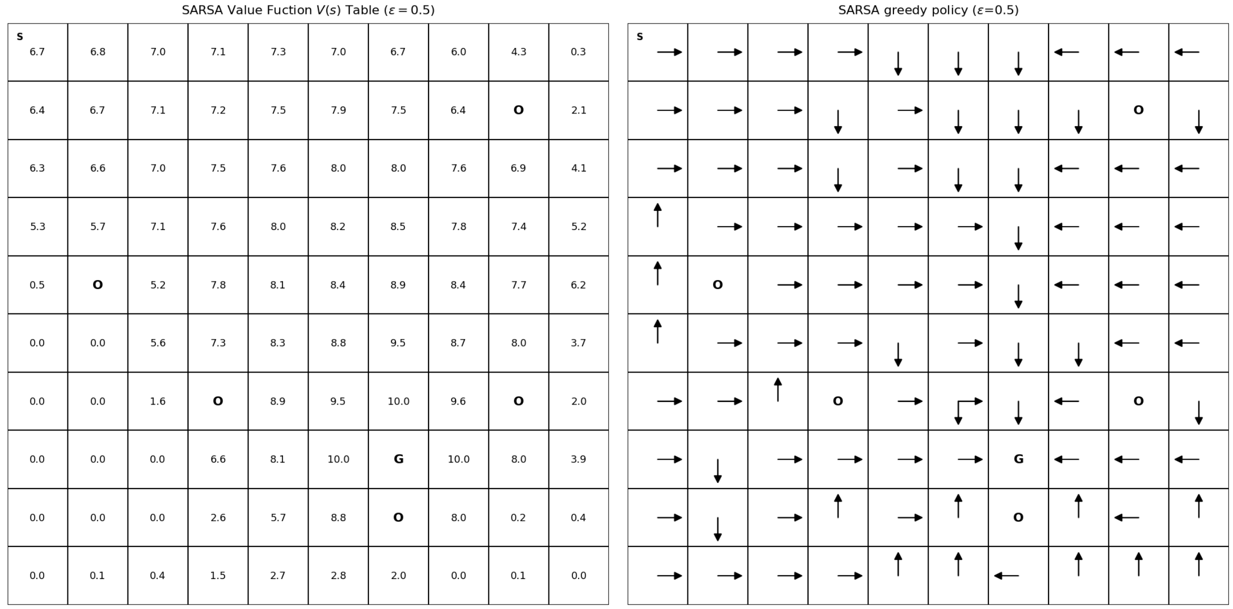
SARSA Value Fuction $V(s)$ Table ($\varepsilon = 0.1$)

S 8.5	8.6	8.7	8.9	8.9	8.5	2.2	0.0	0.0	-0.1
8.3	8.4	8.8	9.0	9.2	9.3	5.7	0.6	O	0.0
3.8	3.9	8.0	9.0	9.1	9.4	9.2	0.4	0.0	0.0
0.0	0.0	0.0	8.5	9.3	9.5	9.3	3.6	0.8	0.0
0.0	O	0.0	2.4	8.0	9.6	9.4	0.0	0.1	0.0
0.0	0.0	0.0	1.9	9.6	9.7	9.9	8.2	1.3	0.0
0.0	0.0	0.0	O	9.6	9.9	10.0	9.9	O	0.0
0.0	0.0	0.0	0.0	6.0	10.0	G	10.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	2.5	O	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

SARSA greedy policy (ε=0.1)

S →	→	→	↓	↓	↓	←	←	←	←↓
↑	→	→	→	→	↓	←	←	O	↓
↑	↑	→	→	→	↓	←	←	↓	↑↓←
↑	↑	→	→	→	↓	←	←	←	←
↑↓	O	→	↑	↓	↓	↓	←	↓	←
↑↓→	↓←→	→	→	→	→	↓	←	←	←
↑↓→	↑↓←→	↑↓←	O	→	→	↓	←	O	↑↓
↑↓→	↑↓←→	↑↓←→	→	→	→	G	←	←	↑↓←
↑↓→	↑↓←→	↑↓←→	↑↓←→	↑	↑	O	↑	↑↓←→	↑↓←
↑→	↑←→	↑←→	↑←→	↑←→	↑	←→	↑←→	↑←→	↑←

Figure 2: State-value function and greedy policy learned by SARSA with $\epsilon = 0.1$.



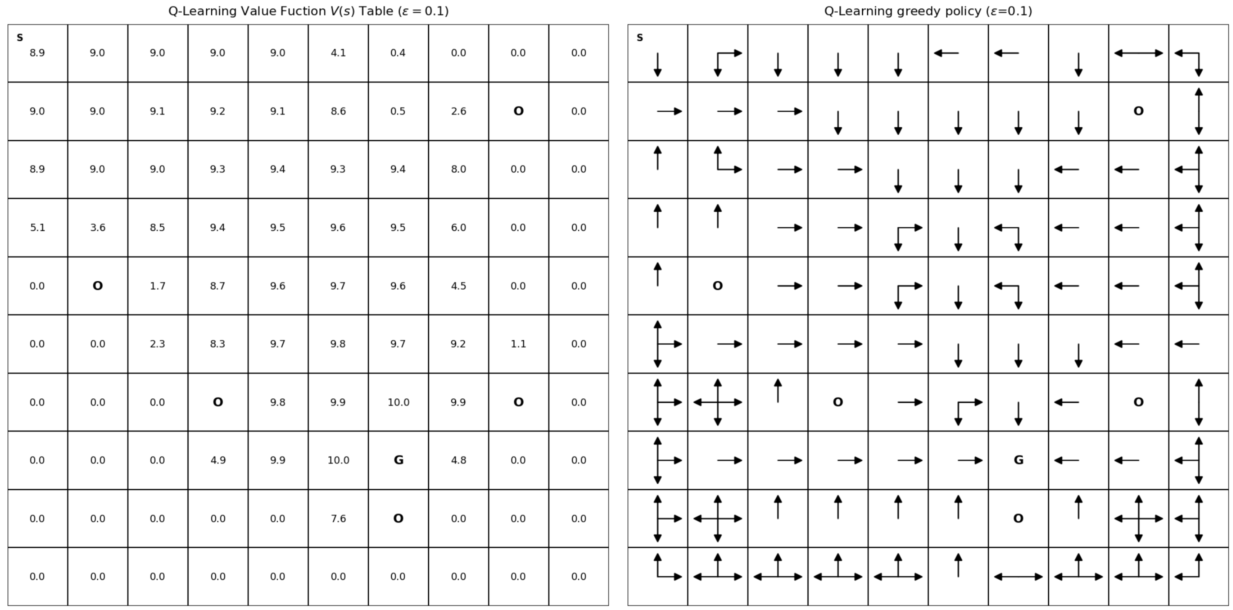
SARSA Value Fuction $V(s)$ Table ($\varepsilon = 0.5$)

S 6.7	6.8	7.0	7.1	7.3	7.0	6.7	6.0	4.3	0.3
6.4	6.7	7.1	7.2	7.5	7.9	7.5	6.4	O	2.1
6.3	6.6	7.0	7.5	7.6	8.0	8.0	7.6	6.9	4.1
5.3	5.7	7.1	7.6	8.0	8.2	8.5	7.8	7.4	5.2
0.5	O	5.2	7.8	8.1	8.4	8.9	8.4	7.7	6.2
0.0	0.0	5.6	7.3	8.3	8.8	9.5	8.7	8.0	3.7
0.0	0.0	1.6	O	8.9	9.5	10.0	9.6	O	2.0
0.0	0.0	0.0	6.6	8.1	10.0	G	10.0	8.0	3.9
0.0	0.0	0.0	2.6	5.7	8.8	O	8.0	0.2	0.4
0.0	0.1	0.4	1.5	2.7	2.8	2.0	0.0	0.1	0.0

SARSA greedy policy (ε=0.5)

S →	→	→	→	↓	↓	↓	←	←	←
→	→	→	↓	→	↓	↓	↓	O	↓
→	→	→	↓	→	↓	↓	←	←	←
↑	→	→	→	→	→	↓	←	←	←
↑	O	→	→	→	→	↓	←	←	←
↑	→	→	→	↓	→	↓	↓	←	←
→	→	↑	O	→	↓→	↓	←	O	↓
→	↓	→	→	→	→	G	←	←	←
→	↓	→	↑	→	↑	O	↑	←	↑
→	→	→	→	↑	↑	←	↑	↑	↑

Figure 3: State-value function and greedy policy learned by SARSA with $\epsilon = 0.5$.

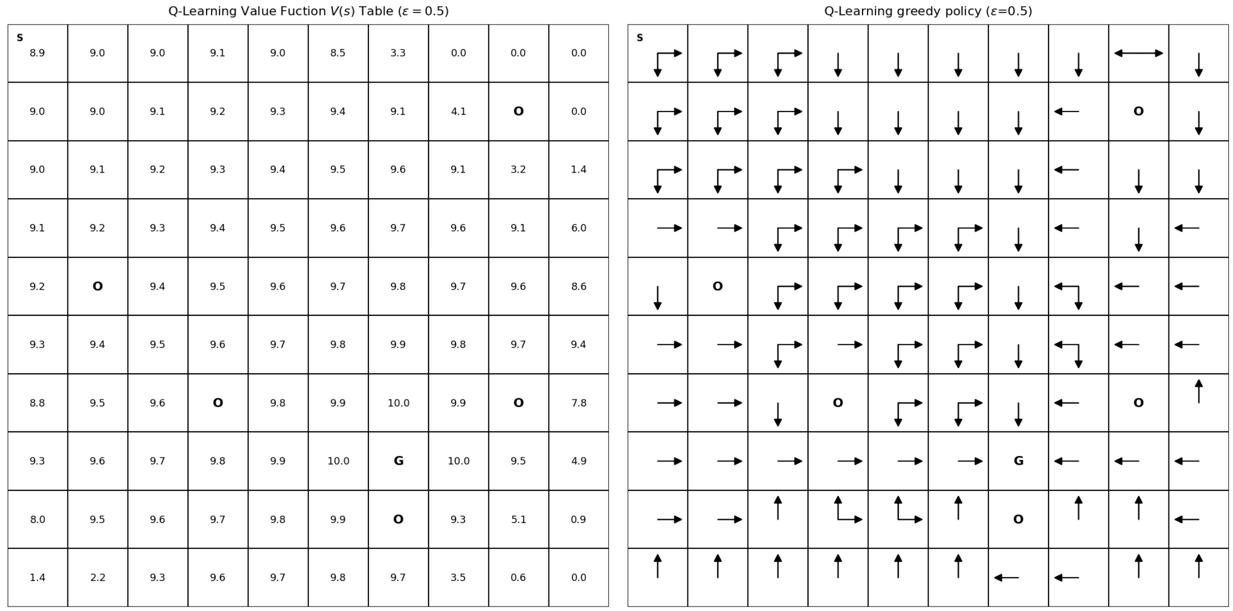


Q-Learning Value Fuction $V(s)$ Table ($\varepsilon = 0.1$)

S 8.9	9.0	9.0	9.0	9.0	4.1	0.4	0.0	0.0	0.0
9.0	9.0	9.1	9.2	9.1	8.6	0.5	2.6	**O**	0.0
8.9	9.0	9.0	9.3	9.4	9.3	9.4	8.0	0.0	0.0
5.1	3.6	8.5	9.4	9.5	9.6	9.5	6.0	0.0	0.0
0.0	**O**	1.7	8.7	9.6	9.7	9.6	4.5	0.0	0.0
0.0	0.0	2.3	8.3	9.7	9.8	9.7	9.2	1.1	0.0
0.0	0.0	0.0	**O**	9.8	9.9	10.0	9.9	**O**	0.0
0.0	0.0	0.0	4.9	9.9	10.0	**G**	4.8	0.0	0.0
0.0	0.0	0.0	0.0	0.0	7.6	**O**	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Q-Learning greedy policy (ε=0.1)

Figure 4: State-value function and greedy policy learned by Q-Learning with $\epsilon = 0.1$.



Q-Learning Value Fuction $V(s)$ Table ($\varepsilon = 0.5$)

S 8.9	9.0	9.0	9.1	9.0	8.5	3.3	0.0	0.0	0.0
9.0	9.0	9.1	9.2	9.3	9.4	9.1	4.1	**O**	0.0
9.0	9.1	9.2	9.3	9.4	9.5	9.6	9.1	3.2	1.4
9.1	9.2	9.3	9.4	9.5	9.6	9.7	9.6	9.1	6.0
9.2	**O**	9.4	9.5	9.6	9.7	9.8	9.7	9.6	8.6
9.3	9.4	9.5	9.6	9.7	9.8	9.9	9.8	9.7	9.4
8.8	9.5	9.6	**O**	9.8	9.9	10.0	9.9	**O**	7.8
9.3	9.6	9.7	9.8	9.9	10.0	**G**	10.0	9.5	4.9
8.0	9.5	9.6	9.7	9.8	9.9	**O**	9.3	5.1	0.9
1.4	2.2	9.3	9.6	9.7	9.8	9.7	3.5	0.6	0.0

Q-Learning greedy policy (ε=0.5)

Figure 5: State-value function and greedy policy learned by Q-Learning with $\epsilon = 0.5$.