Week 3

Policy

A **policy** π is a mapping from states to probabilities of selecting each possible action:

$$\pi(a \mid s) = \Pr(A_t = a \mid S_t = s)$$

Type	Description
Deterministic	$\pi(s) = a$ — always selects the same action in state s
Stochastic	$\pi(a \mid s)$ — probability distribution over actions

Value Functions

State-Value Function

$$v_\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s]$$

- Expected return starting from state s and following policy π
- Measures how good state s is **under policy** π

Action-Value Function

$$q_\pi(s, a) = \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a]$$

- Expected return starting from state s, taking action a, then following policy π
- Measures how good taking action a in state s is **under policy** π

Relationship Between v_π and q_π

From	To	Formula
q_π	v_π	$v_\pi(s) = \sum_a \pi(a \mid s)\, q_\pi(s, a)$
v_π	q_π	$q_\pi(s, a) = \sum_{s', r} p(s', r \mid s, a)\, [r + \gamma v_\pi(s')]$

Here r denotes possible values of the random reward R_{t+1}.

Value Functions Satisfy Recursive Relationships

Value functions can be expressed recursively — the value of the current state depends on the values of successor states. This recursive structure arises from the definition of return:

$$G_t = R_{t+1} + \gamma G_{t+1}$$

Since the return G_t is defined as the immediate reward plus the discounted future return, we can write:

$$v_\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s] = \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s]$$

Markov Property Enables Recursive Closure

The key step from G_{t+1} to $v_\pi(S_{t+1})$ relies on the **Markov property**: the distribution of G_{t+1} only depends on the next state S_{t+1}, not on earlier history.

$$\mathbb{E}_\pi[G_{t+1} \mid S_t = s, A_t = a, S_{t+1} = s'] = \mathbb{E}_\pi[G_{t+1} \mid S_{t+1} = s'] = v_\pi(s')$$

This allows us to substitute G_{t+1} with $v_\pi(S_{t+1})$, yielding the **recursive form**:

$$v_\pi(s) = \mathbb{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s]$$

The Key Compression: Infinite Future → Single Scalar

This is the **mathematical foundation of reinforcement learning**:

$$\underbrace{G_{t+1}}_{\substack{\text{infinite future} \\ \text{random variable}}} \xrightarrow[]{\substack{\text{conditional expectation} \\ \text{(via Markov property)}}} \underbrace{v_\pi(S_{t+1})}_{\substack{\text{expected return} \\ \text{(scalar function)}}}$$

More precisely: $\mathbb{E}_\pi[G_{t+1} \mid S_{t+1} = s'] = v_\pi(s')$

Aspect	G_{t+1}	$v_\pi(S_{t+1})$
Nature	Random variable (sum of infinite rewards)	Scalar function = **expected** return
Depends on	Entire future trajectory	Only current state S_{t+1}
Computational	Impossible to compute directly	Can be estimated recursively
Role	Definition of return	**Compression** via conditional expectation

> **Why this compression is profound**: Without the Markov property, $\mathbb{E}_\pi[G_{t+1} \mid S_t, A_t, S_{t+1}, R_{t+1}]$ would depend on the entire history, making recursive computation impossible. The Markov property enables us to **discard all past conditions** and keep only S_{t+1}.

This compression is what makes the following possible:

- **Dynamic Programming**: Recursively solve for value functions
- **Temporal-Difference Learning**: Learn from incomplete episodes
- **Generalization**: Approximate value functions with neural networks

> The value of a state can be decomposed into the **immediate reward** received after leaving that state, plus the **discounted value** of the successor state.

Component	Meaning
R_{t+1}	Immediate reward after taking action in state s
γG_{t+1}	Discounted future return from successor state s'
$\gamma v_\pi(S_{t+1})$	Discounted value of successor state (by Markov property)

This recursive property is fundamental because:

1. **Enables bootstrapping**: We can estimate the value of a state using estimates of successor states, without waiting for the episode to end
2. **Forms the basis of Bellman equations**: The recursive relationship is formalized into equations that relate values across states
3. **Powers DP and TD methods**: Dynamic programming and temporal-difference learning exploit this structure for efficient computation

Bellman Equations

Bellman Equation for v_π

$$v_\pi(s) = \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a) \left[r + \gamma v_\pi(s') \right]$$

The value of a state equals the expected immediate reward plus the discounted value of the next state.

Derivation:

Starting point: $v_\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s]$ and $G_t = R_{t+1} + \gamma G_{t+1}$

The Three-Layer Structure of Randomness

To compute $\mathbb{E}_\pi[G_t \mid S_t = s]$, we must average over **three layers of randomness**:

```
┌──────────────────────────────────────────────────────────┐
│ Layer 3: Future trajectory -> G_{t+1}                    │
│ (infinite horizon, compressed to v_pi(s'))               │
└──────────────────────────────────────────────────────────┘
                           ^
                           |
┌──────────────────────────────────────────────────────────┐
│ Layer 2: (S_{t+1}, R_{t+1}) ~ p(.|s,a)                   │
│ (environment dynamics)                                   │
└──────────────────────────────────────────────────────────┘
                           ^
                           |
┌──────────────────────────────────────────────────────────┐
│ Layer 1: A_t ~ pi(.|s)                                   │
│ (policy randomness)                                      │
└──────────────────────────────────────────────────────────┘
```

Each application of the **law of total expectation** ”peels off” one layer.

Layer	Random variable	Distribution	What we average over
1	Action A_t	$\pi(a \mid s)$	Policy's choice
2	Next state/reward (S_{t+1}, R_{t+1})	$p(s', r \mid s, a)$	Environment's response
3	Future return G_{t+1}	Trajectory distribution given $S_{t+1} = s'$	**Compressed** by Markov property: $\mathbb{E}_\pi[G$

The **Markov property** is the key that enables Layer 3 compression: the infinite future G_{t+1} is reduced to a single scalar $v_\pi(S_{t+1})$ through conditional expectation.

Step 1 — Peel Layer 1: Condition on action A_t (law of total expectation):

$$\begin{aligned} v_\pi(s) &= \sum_a \underbrace{\Pr(A_t = a \mid S_t = s)}_{\pi(a|s)} \cdot \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s, A_t = a] \\ &= \sum_a \pi(a \mid s)\, \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid s, a] \end{aligned}$$

(Shorthand: $\mathbb{E}_\pi[\cdot \mid s, a] \equiv \mathbb{E}_\pi[\cdot \mid S_t = s, A_t = a]$)

Step 2 — Peel Layer 2: Condition on (S_{t+1}, R_{t+1}):

$$\begin{aligned} \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid s, a] &= \sum_{s',r} \underbrace{\Pr(S_{t+1} = s', R_{t+1} = r \mid s, a)}_{p(s',r|s,a)} \\ &\quad \times \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s, A_t = a, S_{t+1} = s', R_{t+1} = r] \\ &= \sum_{s',r} p(s', r \mid s, a)\, [r + \gamma \mathbb{E}_\pi[G_{t+1} \mid S_t = s, A_t = a, S_{t+1} = s', R_{t+1} = r]] \end{aligned}$$

(Since $R_{t+1} = r$ is fixed, it comes out of the expectation as r)

Step 3 — Layer 3 Compression: Apply Markov property

By the Markov property, G_{t+1} depends only on $S_{t+1} = s'$, **not on earlier history** $(S_t = s, A_t = a, R_{t+1} = r)$:

$$\mathbb{E}_\pi[G_{t+1} \mid S_t = s, A_t = a, S_{t+1} = s', R_{t+1} = r] = \mathbb{E}_\pi[G_{t+1} \mid S_{t+1} = s']$$

This is the key compression: We can **delete all past conditions** and keep only S_{t+1}.

By the **definition of state-value function** (at time $t + 1$, by time-homogeneity):

$$\mathbb{E}_\pi[G_{t+1} \mid S_{t+1} = s'] = v_\pi(s')$$

Infinite future G_{t+1} has been **compressed to a single scalar** $v_\pi(s')$.

Step 4 — Combine all steps:

$$\boxed{v_\pi(s) = \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a)\, [r + \gamma v_\pi(s')]}$$

Intuition: The value of state s averages over:

1. **Actions** (weighted by policy π)
2. **Outcomes** (s', r) (weighted by dynamics p)
3. **Immediate reward** r + **discounted future value** $\gamma v_\pi(s')$

Bellman Equation for q_π

$$q_\pi(s, a) = \sum_{s', r} p(s', r \mid s, a) \left[r + \gamma \sum_{a'} \pi(a' \mid s') \, q_\pi(s', a') \right]$$

Derivation:

Starting point: $q_\pi(s, a) = \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a]$ and $G_t = R_{t+1} + \gamma G_{t+1}$

> **Note**: For q_π, the action $A_t = a$ is **already fixed**, so we have only **two layers** of randomness (not three).

The Two-Layer Structure for q_π

```
┌─────────────────────────────────────────────────────────┐
│ Layer 2: Future trajectory -> G_{t+1}                   │
│ (compressed to v_pi(s'), then expanded via policy)      │
└─────────────────────────────────────────────────────────┘
                           ^
                           |
┌─────────────────────────────────────────────────────────┐
│ Layer 1: (S_{t+1}, R_{t+1}) ~ p(.|s,a)                  │
│ (environment dynamics)                                  │
└─────────────────────────────────────────────────────────┘
```

Layer	Random variable	Distribution
1	Next state/reward (S_{t+1}, R_{t+1})	$p(s', r \mid s, a)$
2	Future return G_{t+1}	Compressed to $v_\pi(s')$, then expanded: $v_\pi(s') = \sum_{a'} \pi(a' \mid s') \, q_\pi(s', a')$

Step 1 — Peel Layer 1: Condition on (S_{t+1}, R_{t+1}):

$$\begin{aligned} q_\pi(s, a) &= \sum_{s', r} \underbrace{\Pr(S_{t+1} = s', R_{t+1} = r \mid s, a)}_{p(s', r \mid s, a)} \cdot \mathbb{E}_\pi[G_t \mid s, a, s', r] \\ &= \sum_{s', r} p(s', r \mid s, a) \, \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s, A_t = a, S_{t+1} = s', R_{t+1} = r] \\ &= \sum_{s', r} p(s', r \mid s, a) \, [r + \gamma \mathbb{E}_\pi[G_{t+1} \mid S_t = s, A_t = a, S_{t+1} = s', R_{t+1} = r]] \end{aligned}$$

(Since $R_{t+1} = r$ is fixed, it comes out of the expectation as r)

Step 2 — Layer 2 Compression: Apply Markov property

By the Markov property, G_{t+1} depends only on $S_{t+1} = s'$, **not on earlier history**:

$$\mathbb{E}_\pi[G_{t+1} \mid S_t = s, A_t = a, S_{t+1} = s', R_{t+1} = r] = \mathbb{E}_\pi[G_{t+1} \mid S_{t+1} = s']$$

By the **definition of state-value function** (at time $t + 1$, by time-homogeneity):

$$\mathbb{E}_\pi[G_{t+1} \mid S_{t+1} = s'] = v_\pi(s')$$

> Infinite future G_{t+1} compressed to scalar $v_\pi(s')$.

So we have the intermediate form:

$$q_\pi(s, a) = \sum_{s',r} p(s', r \mid s, a) \, [r + \gamma v_\pi(s')]$$

Step 3 — Expand v_π back to q_π:

Apply law of total expectation to $v_\pi(s') = \mathbb{E}_\pi[G_{t+1} \mid S_{t+1} = s']$ by conditioning on the **next action** A_{t+1}:

$$\begin{aligned} v_\pi(s') &= \sum_{a'} \underbrace{\Pr(A_{t+1} = a' \mid S_{t+1} = s')}_{\pi(a'|s')} \cdot \mathbb{E}_\pi[G_{t+1} \mid S_{t+1} = s', A_{t+1} = a'] \\ &= \sum_{a'} \pi(a' \mid s') \cdot q_\pi(s', a') \end{aligned}$$

(By time-homogeneity, $\mathbb{E}_\pi[G_{t+1} \mid S_{t+1} = s', A_{t+1} = a'] = q_\pi(s', a')$ — the action-value function definition applies at any time step)

> **Compression followed by expansion**: $G_{t+1} \to v_\pi(s') \to \sum \pi(\cdot) q_\pi(\cdot)$

Step 4 — Substitute $v_\pi(s')$ back:

$$\boxed{q_\pi(s, a) = \sum_{s',r} p(s', r \mid s, a) \left[r + \gamma \sum_{a'} \pi(a' \mid s') \, q_\pi(s', a') \right]}$$

Intuition: The value of action a in state s averages over:

1. **Outcomes** (s', r) (weighted by dynamics p)
2. **Immediate reward** r + **discounted value of next state** $\gamma v_\pi(s')$
3. Where $v_\pi(s')$ itself averages over **next actions** (weighted by policy π)

Backup Diagrams

Diagram	Description
State-value backup	Shows how $v_\pi(s)$ depends on $v_\pi(s')$ for successor states
Action-value backup	Shows how $q_\pi(s,a)$ depends on $q_\pi(s',a')$ for successor state-action pairs

- White circle: state node
- Black dot: action node
- Arcs from state nodes represent policy $\pi(a \mid s)$
- Arcs from action nodes represent dynamics $p(s', r \mid s, a)$

Optimal Value Functions

Optimal State-Value Function

$$v_*(s) = \max_\pi v_\pi(s), \quad \forall s \in \mathcal{S}$$

Optimal Action-Value Function

$$q_*(s,a) = \max_\pi q_\pi(s,a), \quad \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$$

Relationship

$$v_*(s) = \max_a q_*(s,a)$$

$$q_*(s,a) = \sum_{s',r} p(s', r \mid s, a)\left[r + \gamma v_*(s')\right]$$

Bellman Optimality Equations

For v_*

$$v_*(s) = \max_a \sum_{s',r} p(s', r \mid s, a)\left[r + \gamma v_*(s')\right]$$

For q_*

$$q_*(s,a) = \sum_{s',r} p(s', r \mid s, a)\left[r + \gamma \max_{a'} q_*(s',a')\right]$$

Optimal Policy

A policy π is optimal if $v_\pi(s) \geq v_{\pi'}(s)$ for all states s and all policies π'.

Property	Description
Existence	At least one optimal policy always exists
Shared value	All optimal policies share the same v_* and q_*
Greedy extraction	Given q_*, optimal policy is $\pi_*(s) = \arg\max_a q_*(s,a)$