# AC

**Actor-Critic (AC)** algorithms are a class of reinforcement learning methods that combine the strengths of policy gradient and value function-based methods. They consist of two main components:

- **Actor:** A policy function that maps states to actions.
- **Critic:** A value function that estimates the expected future reward for being in a given state or taking a particular action.

# Algorithm Steps

## Initialization:

- Initialize the actor policy function `π(a|s, θ)`.
- Initialize the critic value function `V(s)`.
- Set the learning rate `α` and discount factor `γ`.

## Interaction:

- For each episode:
    - Initialize the current state `s`.
    - While the episode is not terminal:
        - Choose an action `a` from the policy `π(a|s, θ)`.
        - Take action `a` and observe the next state `s'` and reward `r`.
        - Update the critic's value function using temporal difference (TD) learning:

          $$V(s) \leftarrow V(s) + \alpha(r + \gamma V(s') - V(s))$$

        - Update the actor's policy using policy gradients:

          $$\nabla\theta J(\theta) = \nabla\theta log\pi(a|s,\theta)(r + \gamma V(s') - V(s))\theta \leftarrow \theta + \alpha\nabla\theta J(\theta)$$

        - Update the current state `s` to `s'`.

# Advantages of AC Algorithms:

- **Efficient Learning:** AC algorithms can learn more efficiently than pure policy gradient or value function-based methods.
- **Reduced Variance:** The critic can provide a more stable estimate of the expected future reward, leading to reduced variance in policy updates.
- **Flexibility:** AC algorithms can be applied to a wide range of problems, including continuous action spaces and complex environments.