Definition and History of Reinforcement Learning

1. Definition

Reinforcement Learning (RL) is a branch of machine learning concerned with **how an agent ought to take actions in an environment** so as to **maximize cumulative reward** .

Formally, RL can be defined as:

> *A computational approach to goal-directed learning and decision making in which an agent learns to achieve goals through interaction with an environment, by receiving evaluative feedback (rewards) about its actions rather than explicit instructions.*

In contrast to **supervised learning** (which uses labeled examples) and **unsupervised learning** (which finds structure in data), RL emphasizes:

- **Sequential decision making** rather than single-step prediction.
- **Delayed rewards —** current actions influence long-term returns.
- **Exploration–exploitation trade-off —** the agent must balance trying new actions vs. exploiting known rewarding ones.

---

2. The Core Framework

RL involves three key components :

Component	Role
Agent	The learner or decision-maker.

Component	Role
Environment	Everything the agent interacts with.
Reward Signal	Numerical feedback indicating the value of an action's outcome.

At each time step t:

- The agent observes a **state** S_t,
- Takes an **action** A_t,
- Receives a **reward** R_{t+1},
- And the environment transitions to a **new state** S_{t+1}.

The goal is to learn a **policy** $\pi(a \mid s)$ that maximizes the **expected return**:

$$G_t = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \right],$$

where γ (0 ≤ γ < 1) discounts future rewards.

---

3. Historical Development

Period	Milestone & Key Idea
1940s – 1950s	**Behaviorist psychology** (Thorndike's *Law of Effect*, 1933; Skinner's *operant conditioning*, 1938) laid the biological foundation — organisms learn behavior reinforced by reward or punishment.
1950s – 1960s	Early computational analogues: **Donald Hebb's rule (1949)** inspired local credit assignment; **Minsky's SNARC (1954)** simulated learning via reward.
1960s – 1970s	**Dynamic Programming (Bellman, 1957)** introduced optimal control through value functions and Bellman equations . **Samuel's Checkers Program (1959)** was an early self-learning system using rewards.
1980s	Formal unification of trial-and-error learning with DP → **temporal-difference (TD) learning** (Sutton 1988). This decade produced **Q-Learning** (Watkins 1989) and **Actor–Critic architectures**.
1990s – 2000s	RL became mathematically mature. Sutton & Barto's *Reinforcement Learning: An Introduction* (1998) standardized terminology and theory.

Period	Milestone & Key Idea
2010s – 2020s	The deep-learning era: **Deep Q-Network (DQN)** by DeepMind (2015) achieved human-level Atari play by combining RL with neural networks. Extensions include **PPO, A3C, SAC, and AlphaZero**.
Today	RL drives **autonomous control**, **game AI**, **robotics**, **resource optimization**, and increasingly serves as a **theoretical framework for agency in LLMs**.

---

4. Philosophical Intuition

RL mirrors **natural intelligence**: humans and animals learn from experience, forming habits through **reward prediction errors —** the same principle found in dopamine-based learning mechanisms.
Modern RL thus lies at the intersection of **neuroscience**, **psychology**, and **control theory**.

---

5. Modern Definition Recap

> **Reinforcement Learning = Goal-directed learning through interaction, driven by reward signals, optimized over time via trial, error, and feedback.**