2. Planning là gì? (Define Planning)

- **Planning** là quá trình **dùng mô hình** của môi trường (model) làm đầu vào, rồi **sinh ra** (hoặc " tưởng tượng ") các bước tương tác—mà không cần tương tác thật với môi trường—để **cập nhật** hoặc **cải thiện** chính sách.
- Kết quả của Planning là một **chính sách được cải thiện** (improved policy), dựa hoàn toàn trên trải nghiệm "giả lập" do mô hình cung cấp.

---

3. Planning giúp cải thiện chính sách thế nào? (How planning is used to improve policies)

1. **Lấy mẫu từ mô hình**
   - Từ mô hình, ta **chọn ngẫu nhiên** một cặp (trạng thái, hành động) (s, a).
   - Gọi mô hình giả lập (sample model) với (s, a), nó trả về một **mẫu** (s', r) – nghĩa là: nếu ở trạng thái s và thực hiện a, ta sẽ đến s' và nhận reward r.
2. **Cập nhật giá trị**
   - Coi (s, a, r, s') vừa có được như một transition thật sự xảy ra, ta dùng **Q-Learning update** (hoặc SARSA, TD, v.v.) để điều chỉnh giá trị hành động $Q(s, a)$.
   - Ví dụ Q-Learning:

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)].$$

3. **Cải thiện chính sách**
   - Sau khi cập nhật Q, ta **lấy greedy** hoặc **ε-greedy** theo Q mới để thu được **chính sách cải tiến** tại trạng thái s.
   - Lặp lại quá trình này nhiều lần, policy ngày càng tốt lên.

Lưu ý: toàn bộ các bước trên diễn ra **mà không cần** agent phải vào môi trường thật—chính là "planning" dựa trên mô hình.

---

4. Random-sample One-step Tabular Q-planning

Đây là một thuật toán planning đơn giản, mà slide mô tả như sau :

1. **Khởi tạo** giá trị $Q(s, a)$ cho mọi s, a (ví dụ đặt 0).
2. **Lặp** cho đến khi đủ số bước planning:

- **Chọn ngẫu nhiên** một cặp (s, a) trong toàn bộ không gian trạng thái–hành động.
- **Sample** từ mô hình: $(s', r) \sim \text{model.sample}(s, a)$.
- **Update** $Q(s, a)$ theo công thức Q-Learning:

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)].$$

- **Cải thiện policy** tại s:

$$\pi(s) \leftarrow \arg\max_{a} Q(s, a).$$

3. **Kết quả**: policy π ngày càng tiệm cận chính sách tối ưu, hoàn toàn dựa trên **trải nghiệm mô phỏng**.

---

5. Quy trình Planning điển hình trong RL (How planning typically works)

Trong một hệ thống RL kết hợp cả **học từ môi trường thật** và **planning**, thường có ba vòng lặp:

1. **Interact (Học từ thế giới thật)**
   - Agent tương tác với môi trường thật: quan sát (s, a, r, s'), cập nhật Q (hoặc V) bằng các thuật toán TD / Monte Carlo.
   - Push transition vào **bộ nhớ mô hình** (model buffer), dùng để planning sau đó.
2. **Plan (Học từ mô hình)**
   - Lấy transition đã lưu (hoặc lấy mẫu mới) từ mô hình, thực hiện một số bước cập nhật **giả lập** (như Random-sample Q-planning).
   - Không cần agent đang chạy—có thể song song hoặc xen kẽ với tương tác thật.
3. **Policy Improvement**
   - Sau mỗi batch cập nhật (từ tương tác thật hoặc planning), policy được cải thiện (greedy/ε-greedy theo Q).
   - Agent dùng policy này cả khi tương tác thật lẫn khi planning.

   Cách này là nền tảng cho **Dyna** architecture (Sutton, 1991), nơi học song song từ dữ liệu thật và dữ liệu giả lập.

---

6. Tóm tắt (Summary)

- **Planning** trong RL: sử dụng **mô hình** (model) để sinh ra trải nghiệm "giả lập", từ đó cập nhật và **cải thiện policy** mà không cần tương tác thật.

- **Random-sample one-step Q-planning**: thuật toán planning cơ bản, lặp ngẫu nhiên chọn (s, a), sample, update Q, rồi greedy hóa policy.
- Trong ứng dụng thực tế, **planning** thường triển khai **song song** với learning from real experience, để tận dụng tối đa cả dữ liệu thật và dữ liệu mô hình, giúp agent học nhanh hơn, hiệu quả mẫu cao hơn.