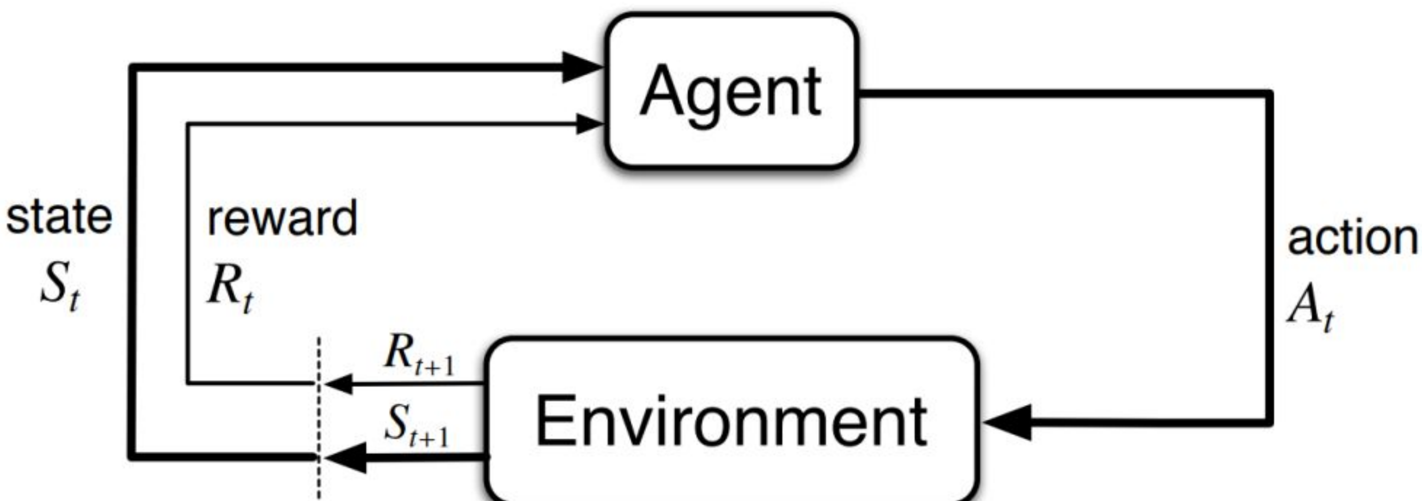
Model Based Reinforcement Learning

B. Ravindran

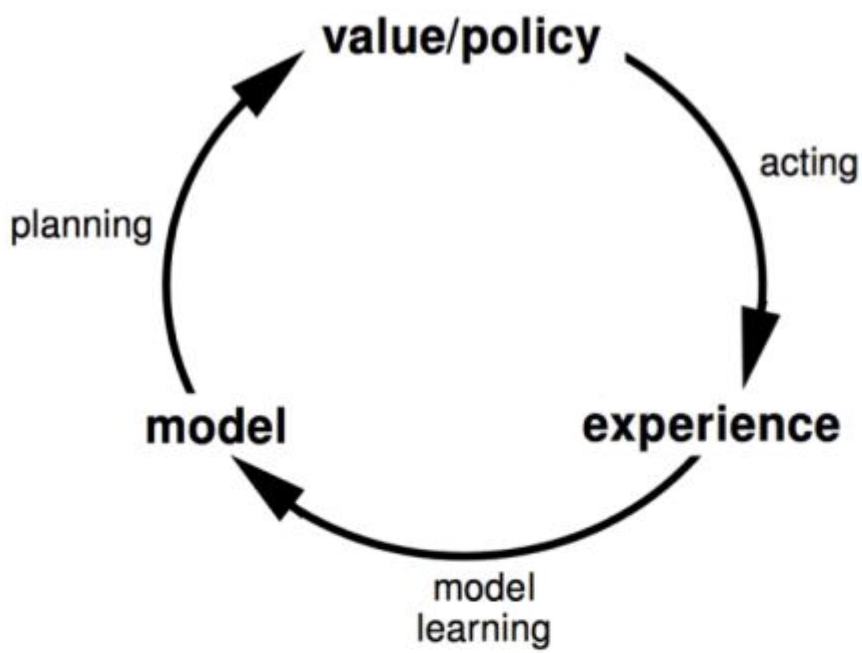
Model-free RL

- No model of the environment.
- **Learn** value function and/or policy directly from experience.
- Experience collected through interaction with the environment.



Model-based RL

- What if we can learn the dynamics of the environment?
- Learn a model of the environment dynamics
- Generate samples using the model.
- Learn/plan using those samples.



Model-based RL

- Advantages:
  - Can efficiently learn model by supervised learning methods.
  - Can reason about model uncertainty.
  - Much more sample-efficient than model-free methods.
  - Transferability and generalization.
- Disadvantages:
  - Additional source of approximation error in model learning.
  - Poor model learning can lead to policies that perform suboptimally in the real environment.

The Model

- Parameterized way of representing an MDP.
- Suppose model is parameterized by η .
- A model can represent state transitions and rewards as follows:

$$S_{t+1} \sim \mathcal{P}_\eta(S_{t+1} \mid S_t, A_t)$$
$$R_{t+1} = \mathcal{R}_\eta(R_{t+1} \mid S_t, A_t)$$

- We typically assume conditional independence between next states and rewards.

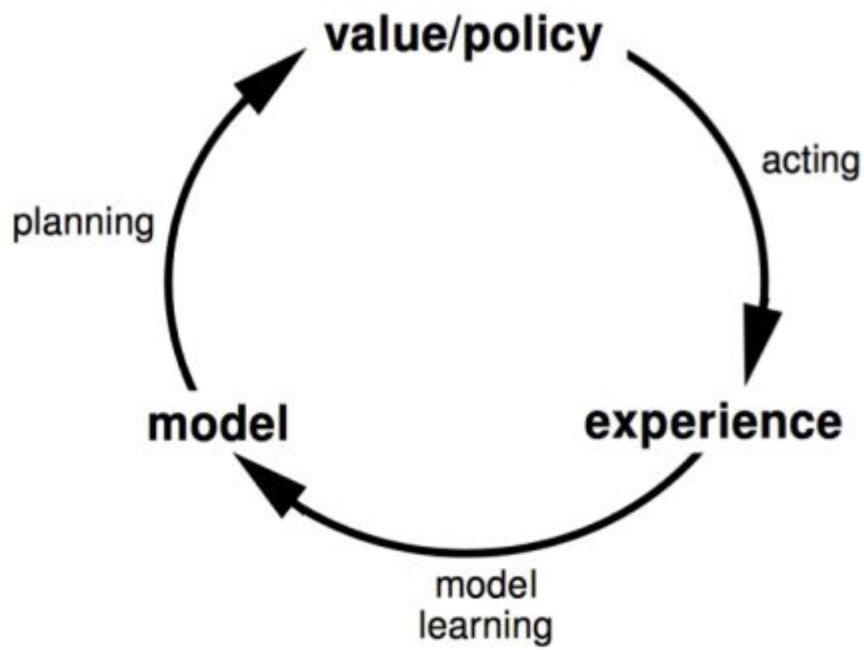
$$\mathbb{P}[S_{t+1}, R_{t+1} \mid S_t, A_t] = \mathbb{P}[S_{t+1} \mid S_t, A_t]\, \mathbb{P}[R_{t+1} \mid S_t, A_t]$$

Learning The Model

- Learnt using experiences collected from the environment.
- Learning the model is a supervised learning problem:

$$
\begin{gathered}
S_1, A_1 \to R_2, S_2 \\
S_2, A_2 \to R_3, S_3 \\
\vdots \\
S_{T-1}, A_{T-1} \to R_T, S_T
\end{gathered}
$$

Planning Using The Model



Planning Using The Model

- A powerful sample-efficient approach to RL.
- Experiences are sampled from the learnt model.

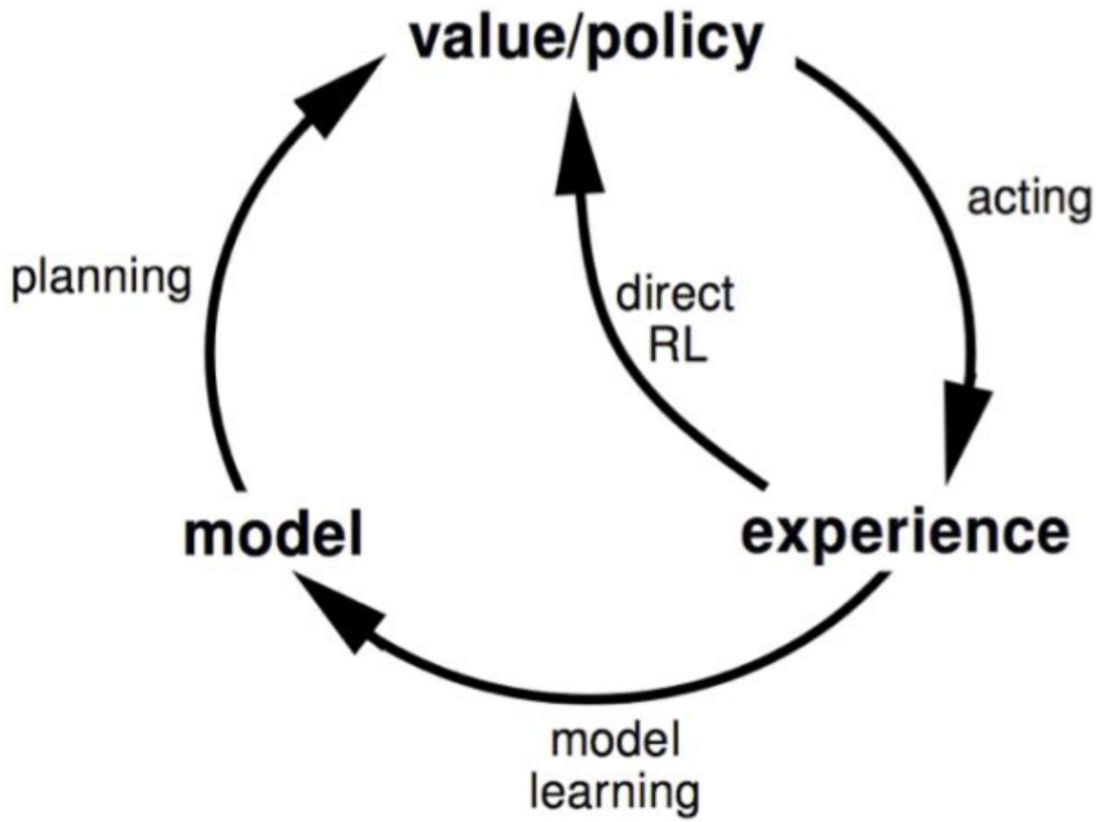
$$S_{t+1} \sim \mathcal{P}_\eta(S_{t+1} \mid S_t, A_t)$$
$$R_{t+1} = \mathcal{R}_\eta(R_{t+1} \mid S_t, A_t)$$

- Apply model-free RL to samples, e.g.:
  - MC-control
  - SARSA
  - Q-learning
  - DQN

Planning Using The Model

1. Interact with the environment.
2. Learn the model.
3. Use the model to generate experiences.
4. Use the **simulated experiences** to train your RL algorithm of choice.
5. Repeat steps 1 to 4 till convergence.

Dyna: Integrating Learning and Planning



Dyna: Integrating Learning and Planning

Sample-based Planning:

- Learn a model from real experience
- Plan using simulated experience.

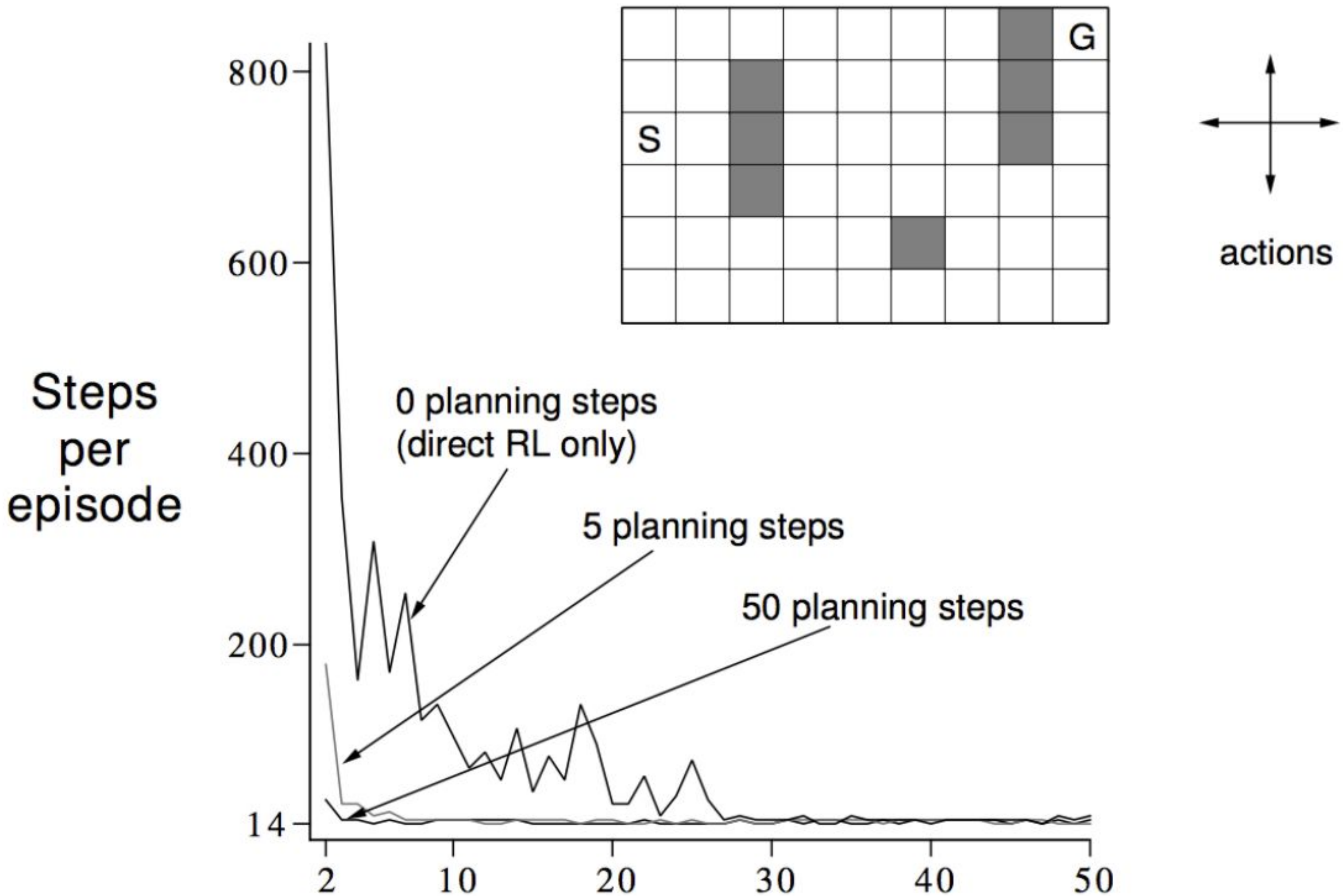
Dyna:

- Learn a model from real experience
- Learn and plan from real and simulated experience.

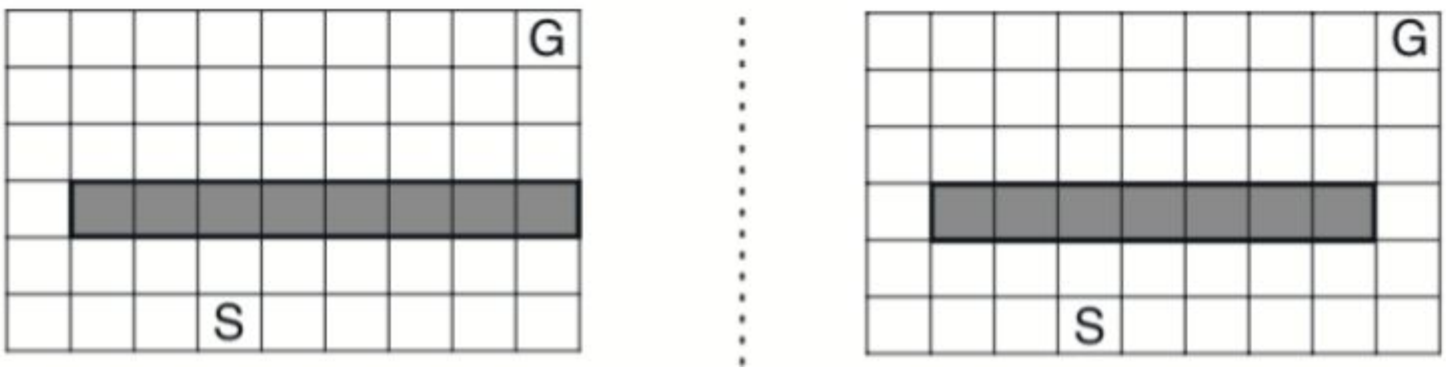
Dyna-Q Learning

Initialize $Q(s, a)$ and $Model(s, a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$
Do forever:
  (a) $S \leftarrow$ current (nonterminal) state
  (b) $A \leftarrow \varepsilon$-greedy(S, Q)
  (c) Execute action A; observe resultant reward, R, and state, S'
  (d) $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma \max_a Q(S', a) - Q(S, A)\big]$
  (e) $Model(S, A) \leftarrow R, S'$ (assuming deterministic environment)
  (f) Repeat n times:
    $S \leftarrow$ random previously observed state
    $A \leftarrow$ random action previously taken in S
    $R, S' \leftarrow Model(S, A)$
    $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma \max_a Q(S', a) - Q(S, A)\big]$

Dyna-Q Learning



Non-stationary Environments: Dyna Q+



- Maze changes dynamically at a certain timestep t.
- At t, a shortcut to **G** will open as shown (right)
- Will a Dyna-Q agent be able to find the optimal solution?
- Agent starts at **S** and needs to reach **G.**

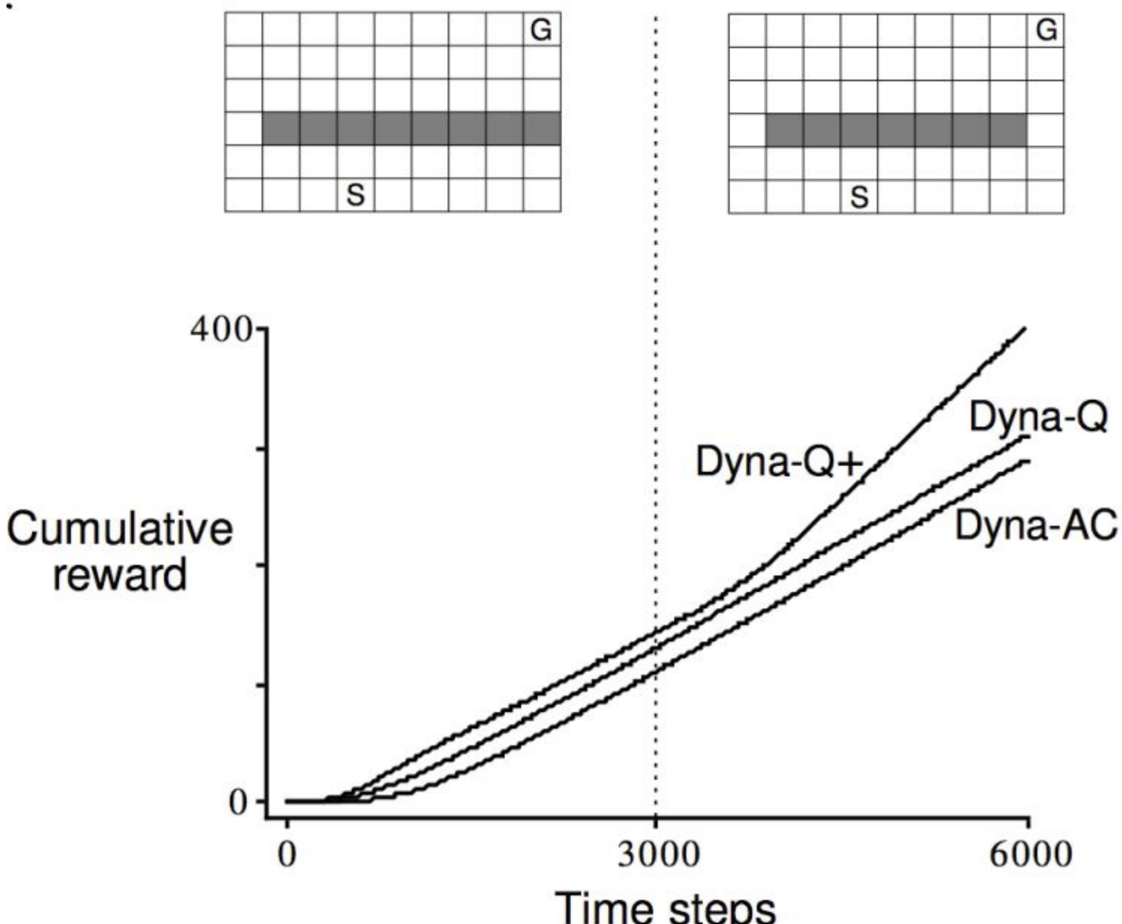
Non-stationary Environments: Dyna Q+

- Solution: Dyna-Q+
- Uses an “exploration bonus”.
- Keeps track of time since each state-action pair was tried in a real interaction with environment.

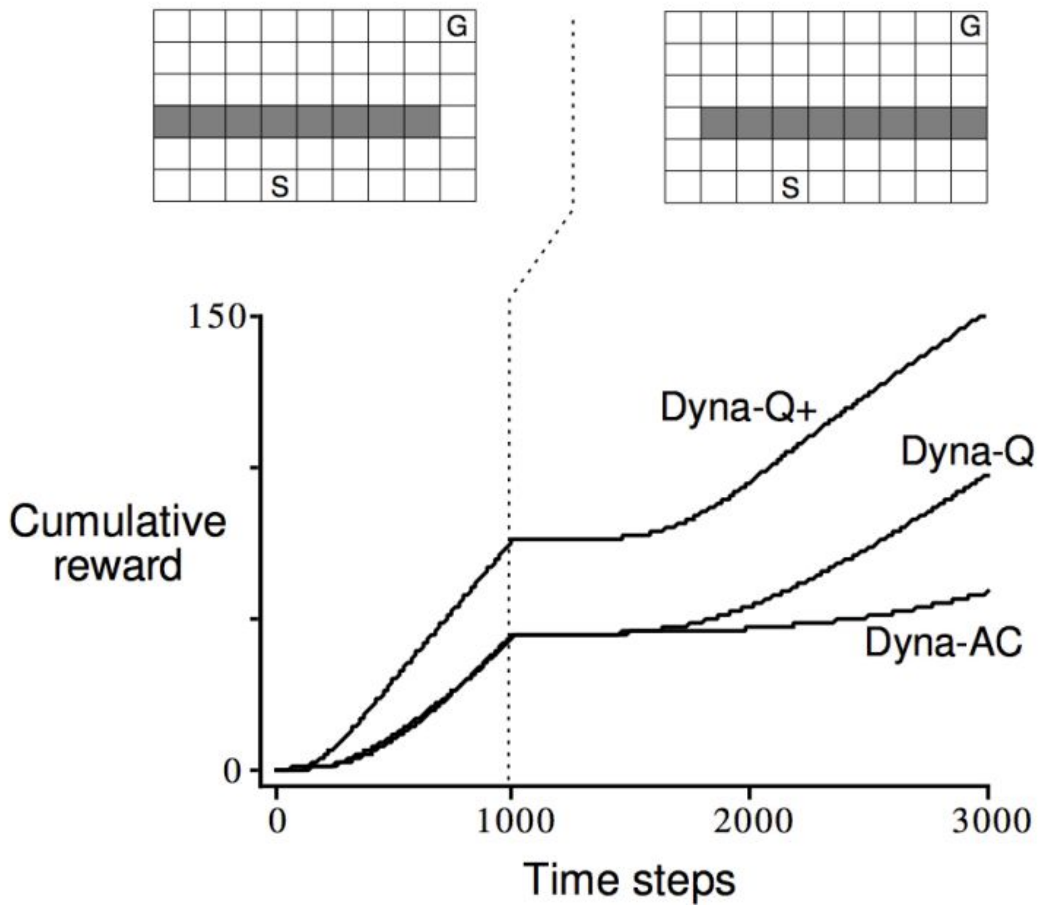
$$r + \kappa\sqrt{n}$$

- An extra reward is added for transitions caused by state-action pairs related to how long ago they were tried: the longer unvisited, the more reward for visiting.

Non-stationary Environments: Dyna Q+

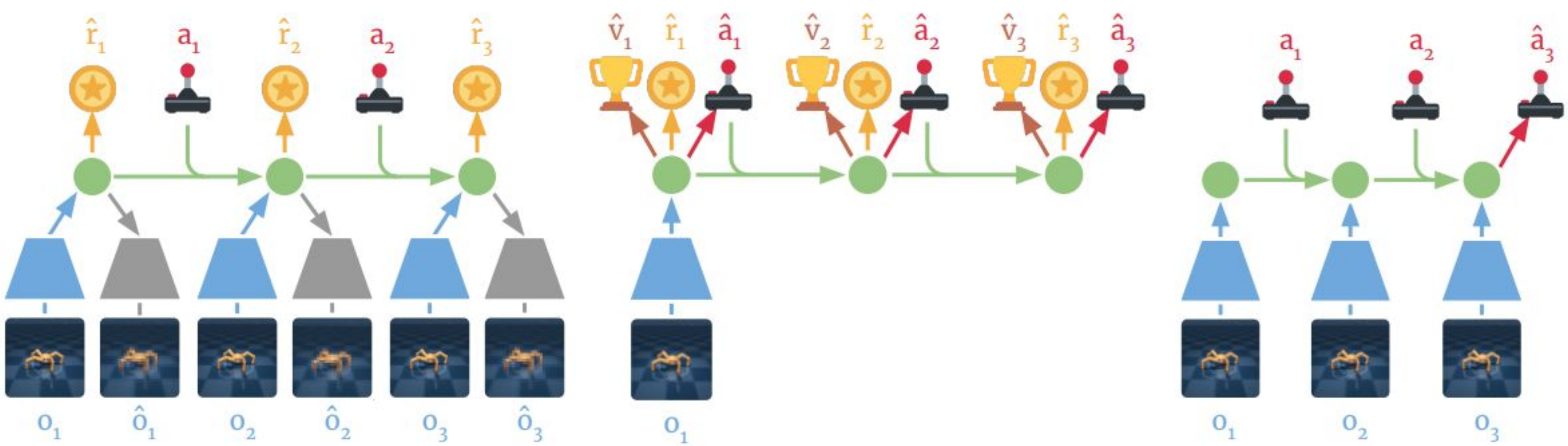


Non-stationary Environments: Dyna Q+



Recent Advances: Latent Variable Models

- Recent advances like Dreamer (Ha et. al.) and Stochastic Latent Actor Critic (Lee et. al.) leverage variational inference to learn latent variable models of environment dynamics.



LEARN ENVIRONMENT DYNAMICS **GENERATE LATENT TRAJECTORIES** **LEARN POLICY**

Recent Advances: Latent Variable Models

- DreamerV2 substantially outperforms previous world models. Moreover, it exceeds top model-free agents within the same compute and sample budget.