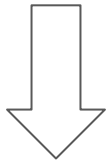
Aprendizado On Policy x Off Policy

Estrutura Básica de Algoritmos de RL

Coleta de Dados



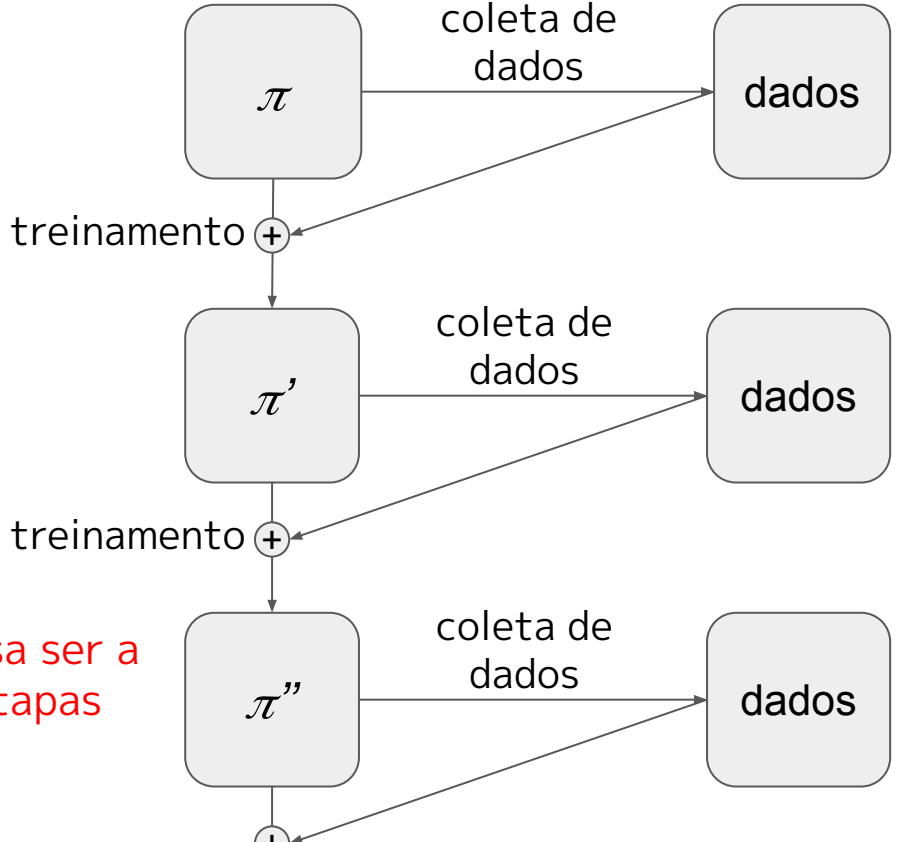
Treinamento

Uma Possibilidade

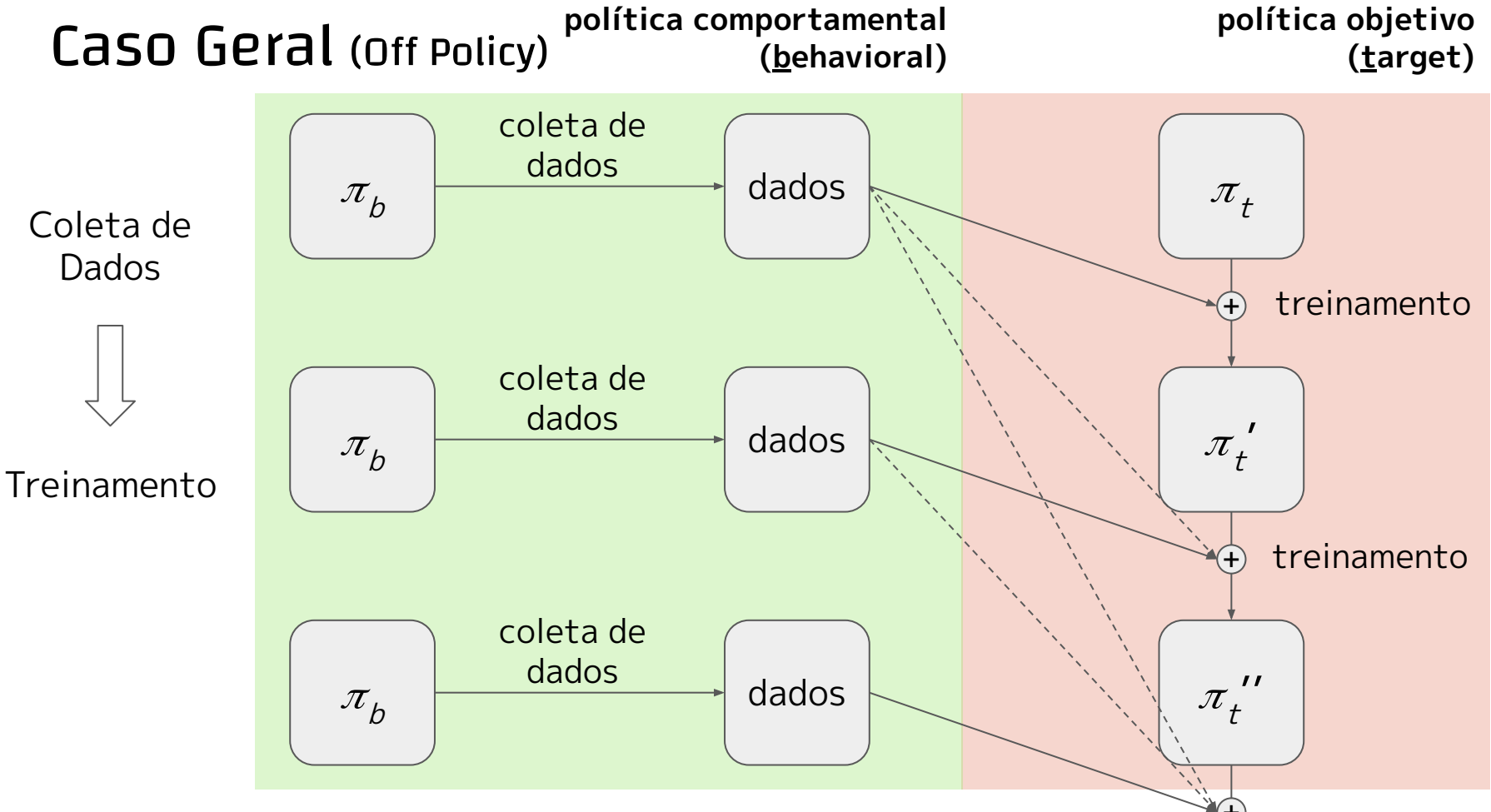
Coleta de Dados

Treinamento

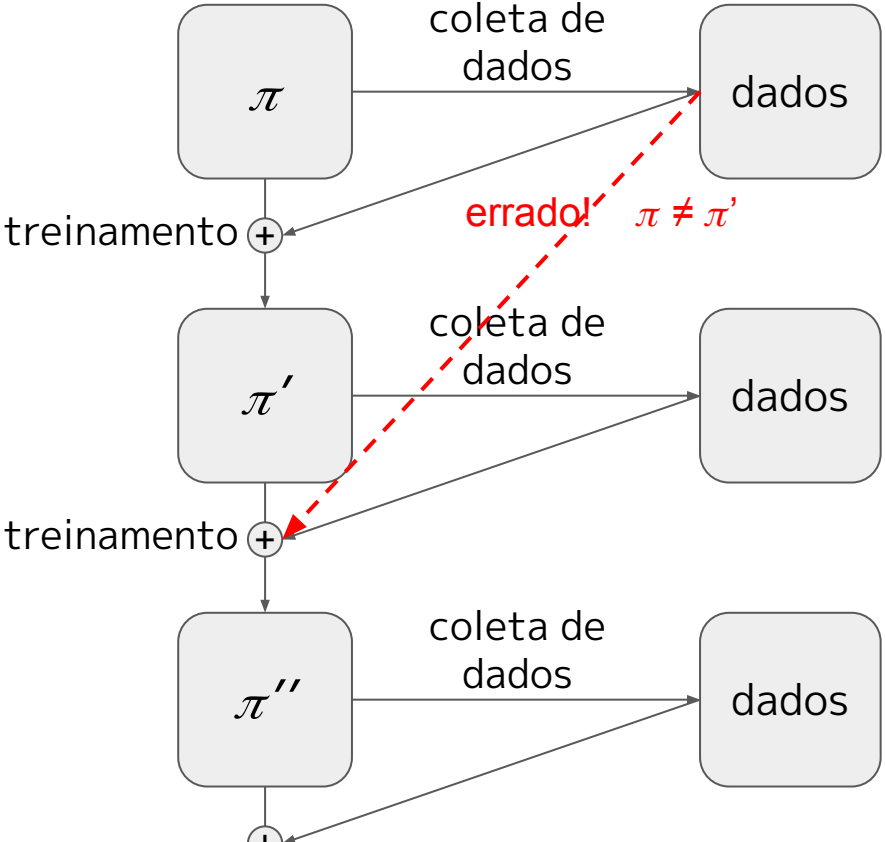
a política não precisa ser a mesma nas duas etapas

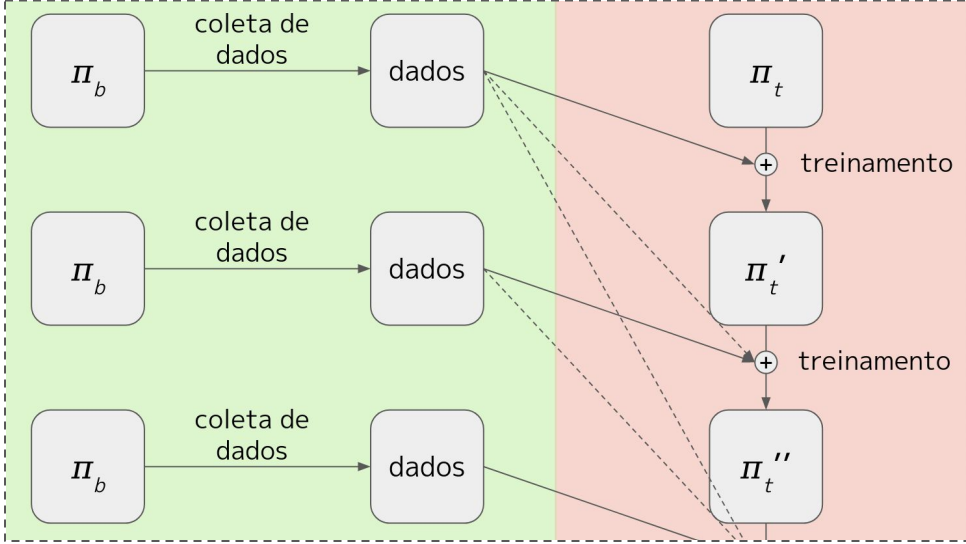


Caso Geral (Off Policy)



On Policy: $\pi_b = \pi_t$

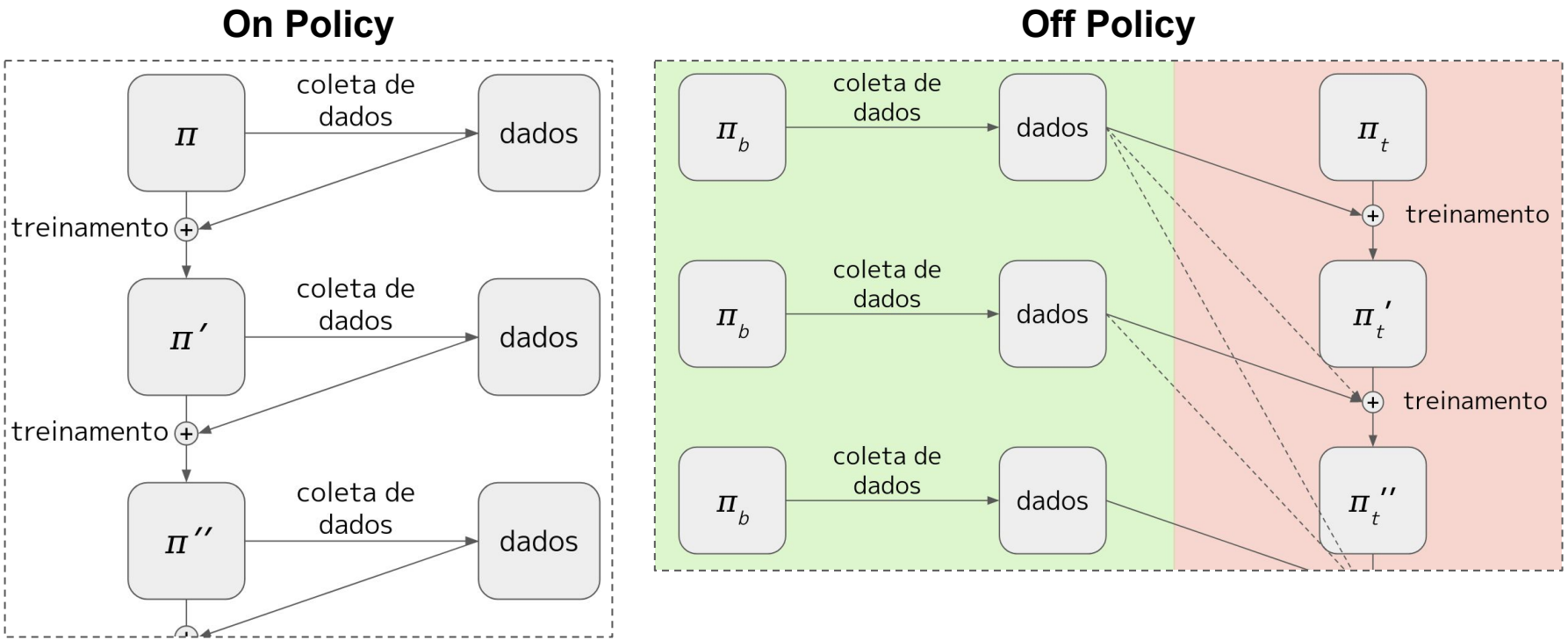




Não podemos utilizar dados:

- antigos (experience replay)
- de outros agentes (e.g. de um agente que pode atuar no mundo real sem risco de acidentes)

Nomenclatura



Resumo

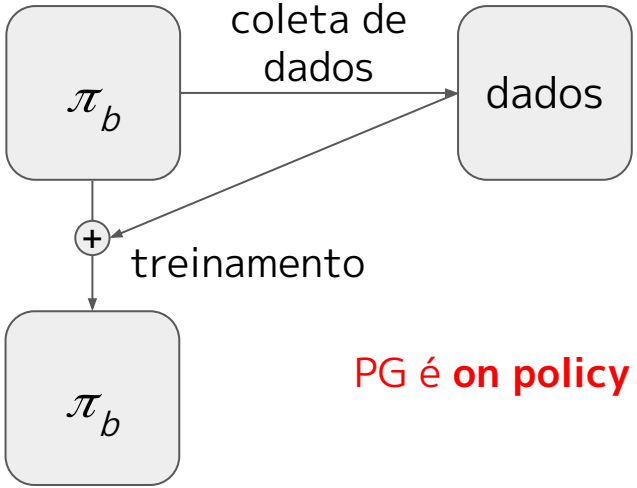
1. **Coleta de dados**
   - Política π_b (behavioral/comportamental), que pode ser:
     - Igual à política target no instante atual: $\pi_b = \pi_t$ } on policy
     - Igual à política target num outro instante: $\pi_b = \pi'_t$ } off policy
     - Algum outro agente *qualquer* com uma política $\pi_b \neq \pi_t$

     (geralmente exige-se que π_b seja conhecido)
2. **Treinamento**
   - Política π_t (target/objetivo)

Exemplo: Policy Gradient

$$\theta \leftarrow \theta + \alpha \cdot \nabla_\theta J(\theta)$$

$$\nabla_\theta J(\theta) = E_\theta\left[Q^{\pi_\theta}(s,a) \cdot \nabla_\theta \log \pi_\theta(a|s) : s, a\right]$$



PG é **on policy**

- A esperança depende da distribuição dos dados (que depende de π_b)
- Logo, o gradiente calculado é específico para π_b

Recap: Equações de Bellman

$$q_\pi(S_t, A_t) = E_\pi\left[R_{t+1} + \gamma \cdot q_\pi(S_{t+1}, A_{t+1}) : S_t = s, A_t = a\right]$$

equação de esperança de Bellman
(usada para estimar o q-valor de uma política)

$$q_*(S_t, A_t) = E\left[R_{t+1} + \gamma \cdot \max_a q_*(S_{t+1}, a) : S_t = s, A_t = a\right]$$

equação de otimalidade de Bellman
(usada para estimar o q-valor da política *ótima*)

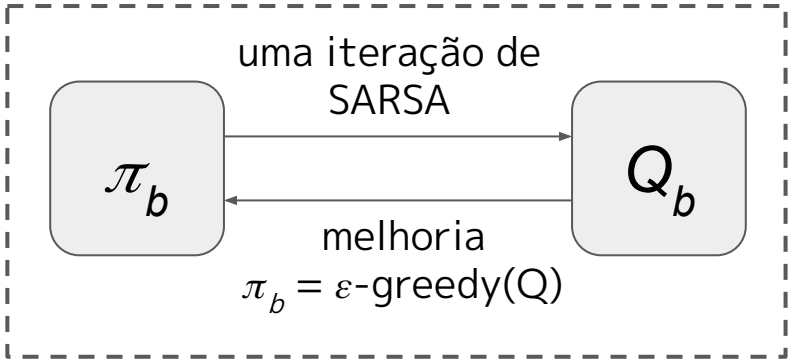
Exemplo: SARSA

equação de esperança

$$q_\pi(S_t, A_t) = E_\pi\left[R_{t+1} + \gamma \cdot q_\pi(S_{t+1}, A_{t+1}) : S_t = s,\ A_t = a\right]$$

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\left[\underbrace{R_{t+1} + \gamma \cdot Q(S_{t+1}, A_{t+1})}_{\text{bootstrap: } q_\pi(S_t, A_t)} - Q(S_t, A_t)\right]$$

SARSA é **on policy**

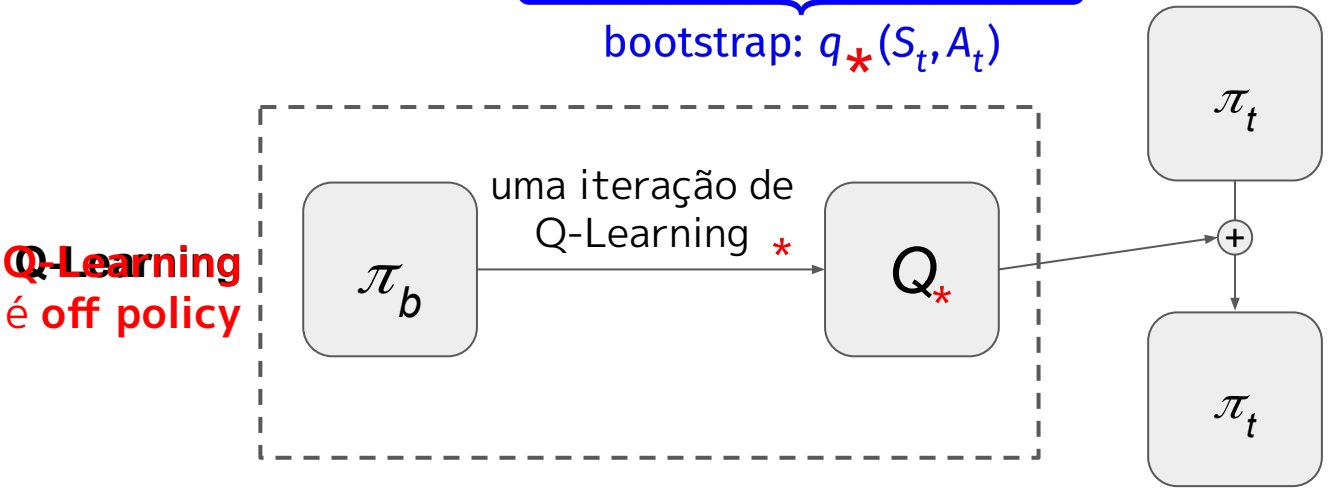


Exemplo: Q-Learning

equação de otimalidade

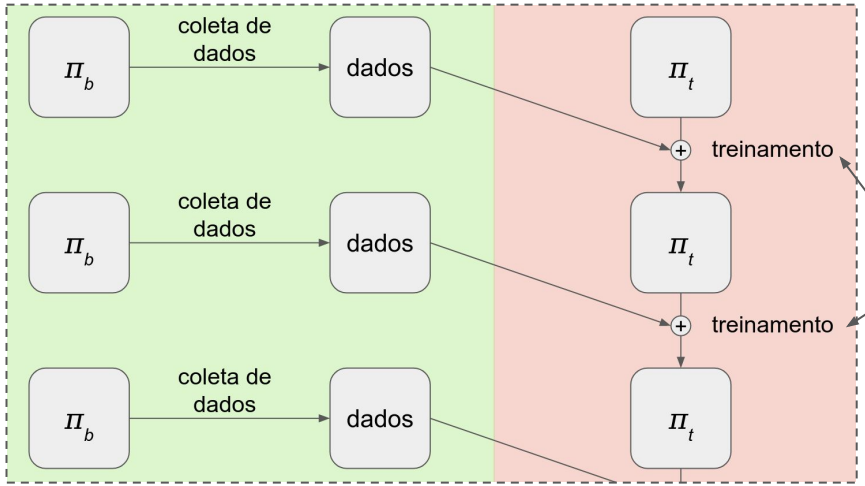
$$q_*(S_t, A_t) = E\left[R_{t+1} + \gamma \cdot \max_a q_*(S_{t+1}, a) : S_t = s, A_t = a\right]$$

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\left[\underbrace{R_{t+1} + \gamma \cdot \max_a q_*(S_{t+1}, a)}_{\text{bootstrap: } q_*(S_t, A_t)} - Q(S_t, A_t)\right]$$



Exemplo: Q-Learning

Q-Learning é **off policy**



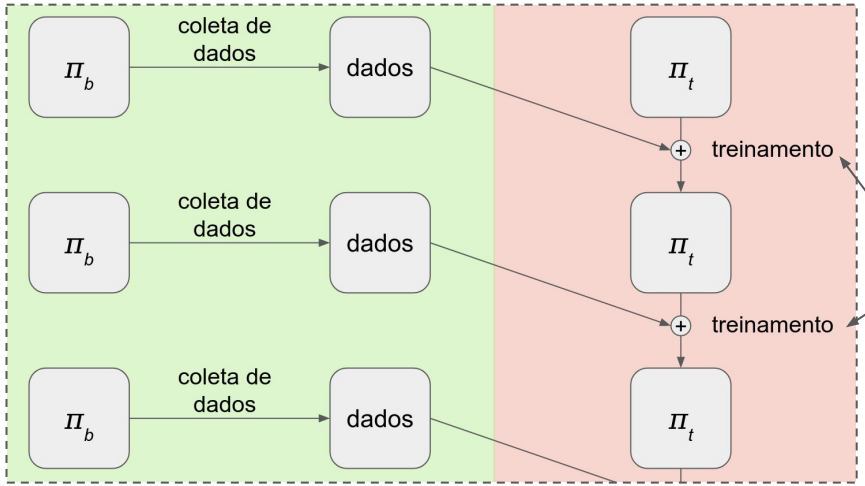
π_t é uma estimativa de π_*
π_b é uma política qualquer

estimativa do valor de π_*

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \right]$$

Exemplo: Q-Learning com ε-greedy

Essa versão de Q-Learning é **on policy**



π_t é uma estimativa de π_*

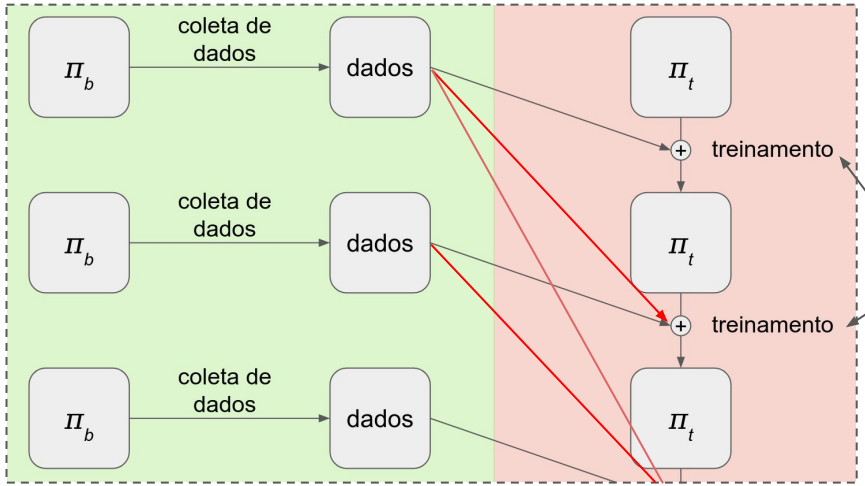
$\pi_b = \pi_t$

estimativa do valor de π_*

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \right]$$

Exemplo: Q-Learning com ε-greedy e replay

Essa versão de Q-Learning é **off policy**



π_t é uma estimativa de π_*

π_b = alguma versão de π_t (não necessariamente a atual)

estimativa do valor de π_*

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \right]$$