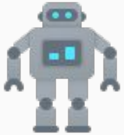
Model Based Autonomous RL

By: Sergio Charles

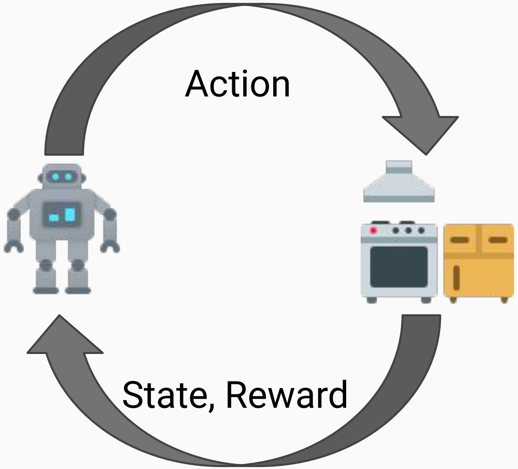
Supervised by: Archit Sharma & Chelsea Finn

Introduction

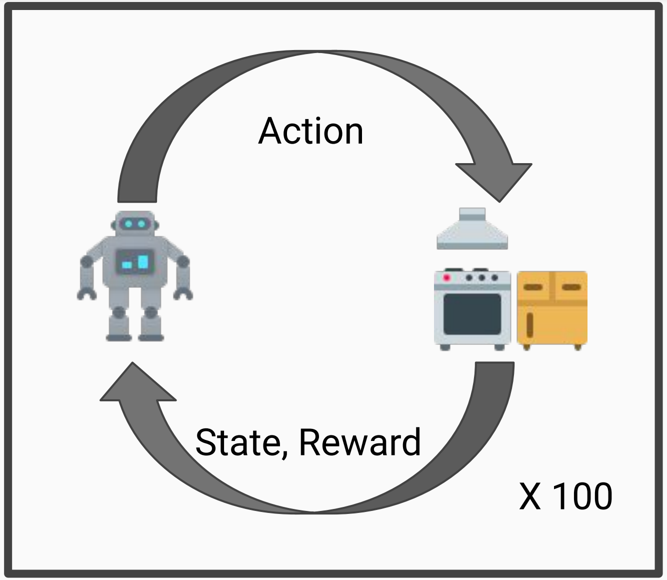
Introduction to Autonomous RL



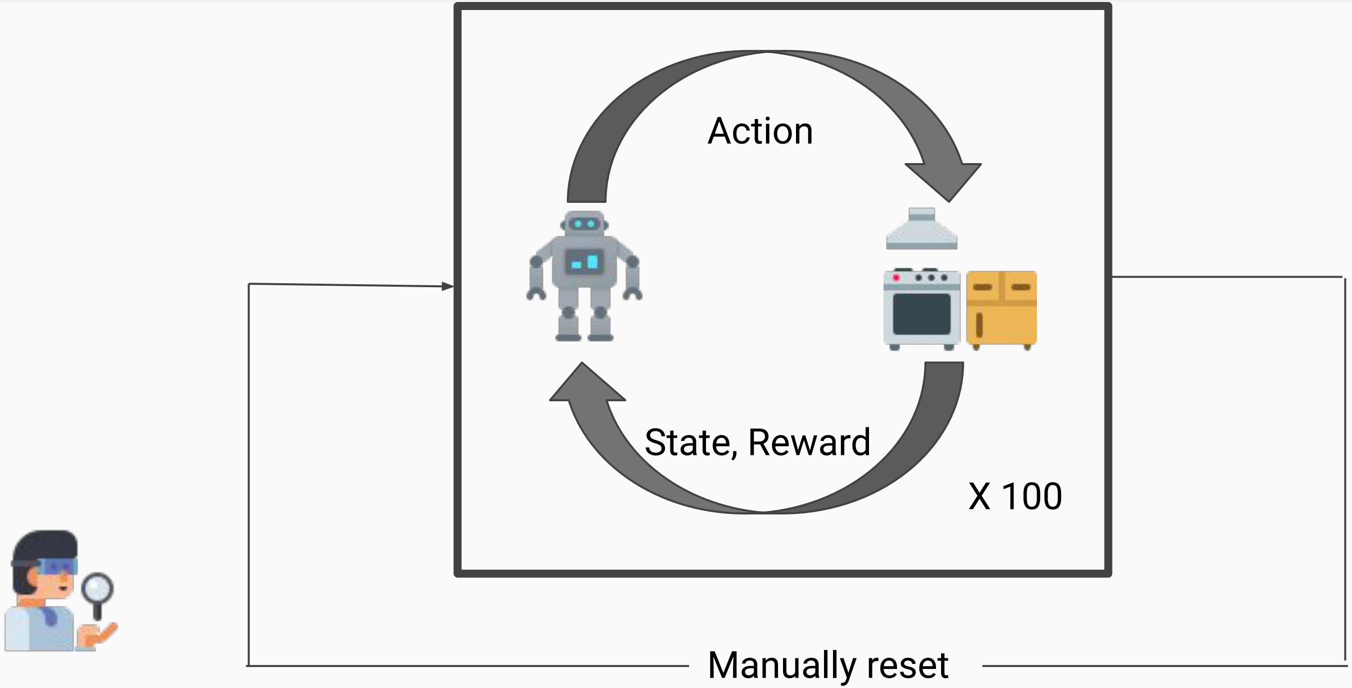
Introduction to Autonomous RL



Introduction to Autonomous RL



Introduction to Autonomous RL



Motivating Question

Question: Embodied agents, e.g. humans and robots, function in a continual, non-episodic world. Why does the research community still develop RL algorithms in episodic settings?

Motivating Question

Question: Embodied agents, e.g. humans and robots, function in a continual, non-episodic world. Why does the research community still develop RL algorithms in episodic settings?

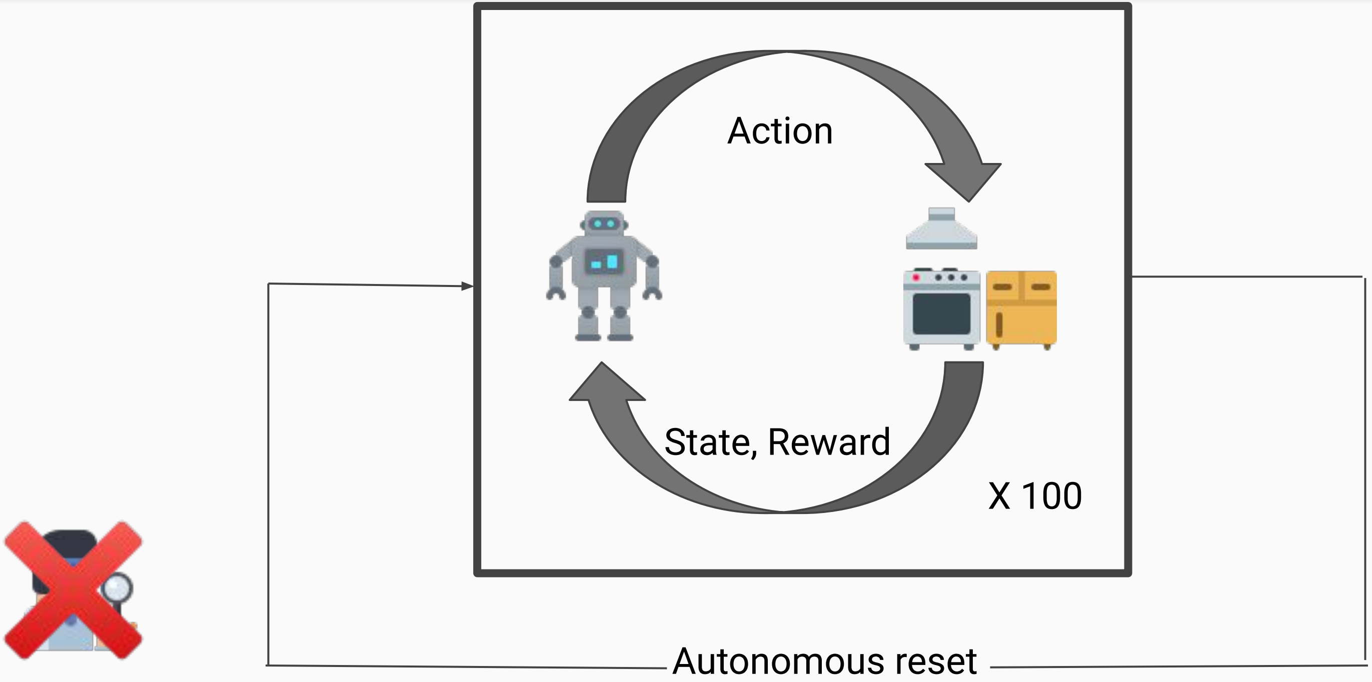
- To build autonomous embodied agents, it is essential to learn continually without human interventions.

Motivating Question

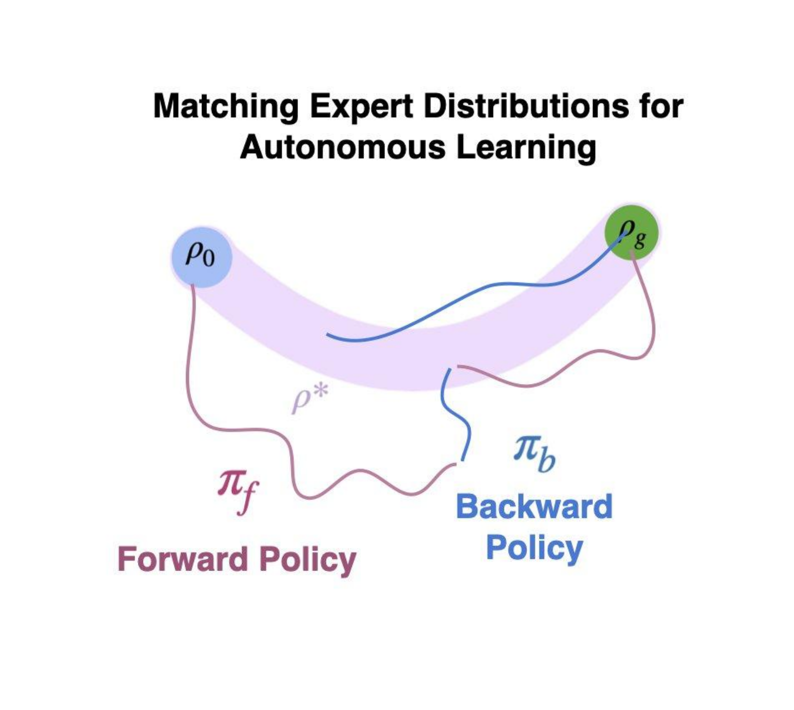
Question: Embodied agents, e.g. humans and robots, function in a continual, non-episodic world. Why does the research community still develop RL algorithms in episodic settings?

- To build autonomous embodied agents, it is essential to learn continually without human interventions.
- Episodic learning requires humans to intervene after every episode, impeding autonomy & scale of learning systems.

Introduction to Autonomous RL



MEDAL



Key idea:

In addition to learning a forward policy π_f to solve the task, learn a backward policy π_b to stay close to the states in the demonstration distribution.

Forward policy objective

Expected discounted sum of rewards:

$$\max_{\pi_f} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$$

Backward policy objective

Minimize Jensen-Shannon divergence between optimal state distribution and the state distribution induced by backward policy:

$$\min_{\pi_b} \mathcal{D}_{\mathrm{JS}}(\rho^b(s) || \rho^*(s))$$

Imitation Learning via Distribution Matching

- We do not require an explicit density under the generative distribution.
- Only require the ability to sample the distribution, allows construction of imitation learning methods

Imitation Learning via Distribution Matching

- To match ρ^b and ρ^*, use a small set of demonstration to learn a state-space classifier $C : S \to [0,1]$.
- Generate states with the backward policy π_b by imitating demonstration states, hence solving the min-max problem:

$$\min_{\pi_b} \max_{C} \mathbb{E}_{s\sim\rho^*}[\log C(s)] + \mathbb{E}_{s\sim\rho^b}[\log(1 - C(s))]$$

Better Starting States

Kakade & Langford, 2002, Corollary 4.5

- Upper bound on the difference between the optimal performance and that of policy π is proportional to:

$$\left\| \frac{\rho^*(s)}{\rho_0(s)} \right\|_\infty$$

- The closer the starting state distribution is to the state distribution of the optimal policy, the faster the policy moves toward the optimal policy π^*.

Motivating a Model Based Approach

Limitations of Model-Free Approaches

- Existing RL algorithms in the non-episodic setting focus on model-free methods (FBRL, R3L, VaPRL, MEDAL).

Limitations of Model-Free Approaches

- Existing RL algorithms in the non-episodic setting focus on model-free methods (FBRL, R3L, VaPRL, MEDAL).
- Model-free methods learn a Q-value function $Q(s, a; \theta)$, e.g. soft-actor acritic, to optimize forward $\pi_f(a|s; \phi_f)$ and backward $\pi_b(a|s; \phi_b)$ controllers.

Limitations of Model-Free Approaches

- Existing RL algorithms in the non-episodic setting focus on model-free methods (FBRL, R3L, VaPRL, MEDAL).
- Model-free methods learn a Q-value function $Q(s, a; \theta)$, e.g. soft-actor acritic, to optimize forward $\pi_f(a|s; \phi_f)$ and backward $\pi_b(a|s; \phi_b)$ controllers.
- Data sharing across forward & backward policies is non-trivial:

Limitations of Model-Free Approaches

- Existing RL algorithms in the non-episodic setting focus on model-free methods (FBRL, R3L, VaPRL, MEDAL).
- Model-free methods learn a Q-value function $Q(s, a; \theta)$, e.g. soft-actor acritic, to optimize forward $\pi_f(a|s; \phi_f)$ and backward $\pi_b(a|s; \phi_b)$ controllers.
- Data sharing across forward & backward policies is non-trivial:
  - Methods like hindsight relabeling for goal-conditioned RL does not work because the two policies are too different.

Limitations of Model-Free Approaches

- Existing RL algorithms in the non-episodic setting focus on model-free methods (FBRL, R3L, VaPRL, MEDAL).
- Model-free methods learn a Q-value function $Q(s, a; \theta)$, e.g. soft-actor acritic, to optimize forward $\pi_f(a|s;\phi_f)$ and backward $\pi_b(a|s;\phi_b)$ controllers.
- Data sharing across forward & backward policies is non-trivial:
  - Methods like hindsight relabeling for goal-conditioned RL does not work because the two policies are too different.
  - Highly sample inefficient.

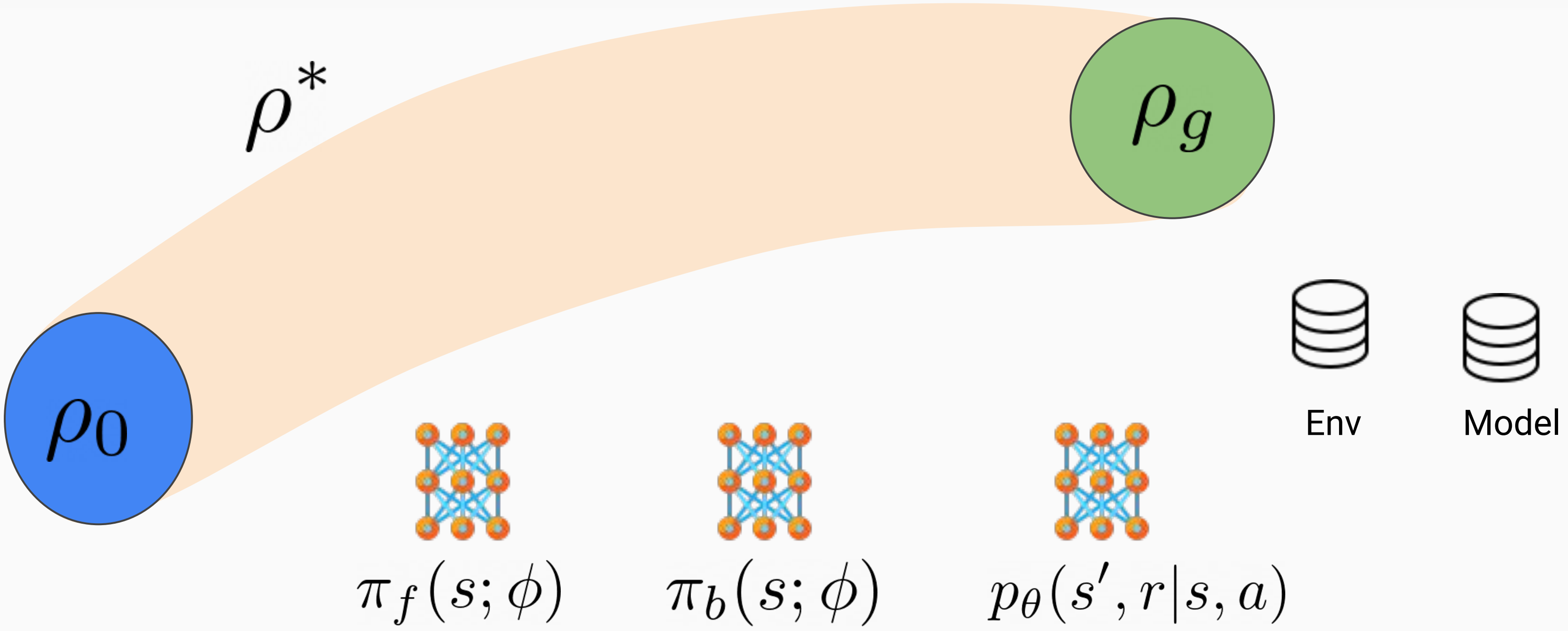
Technical Challenge: Unified Model

- Learn a **unified dynamics model** $p(s', r|s, a; \omega)$ to efficiently construct a forward policy $\pi_f(a|s; \phi_f)$ and backward policy $\pi_b(a|s; \phi_b)$ around the demonstration state distribution ρ^*.
- Use data collected by π_f and π_b to train the same dynamics model for sample efficiency.

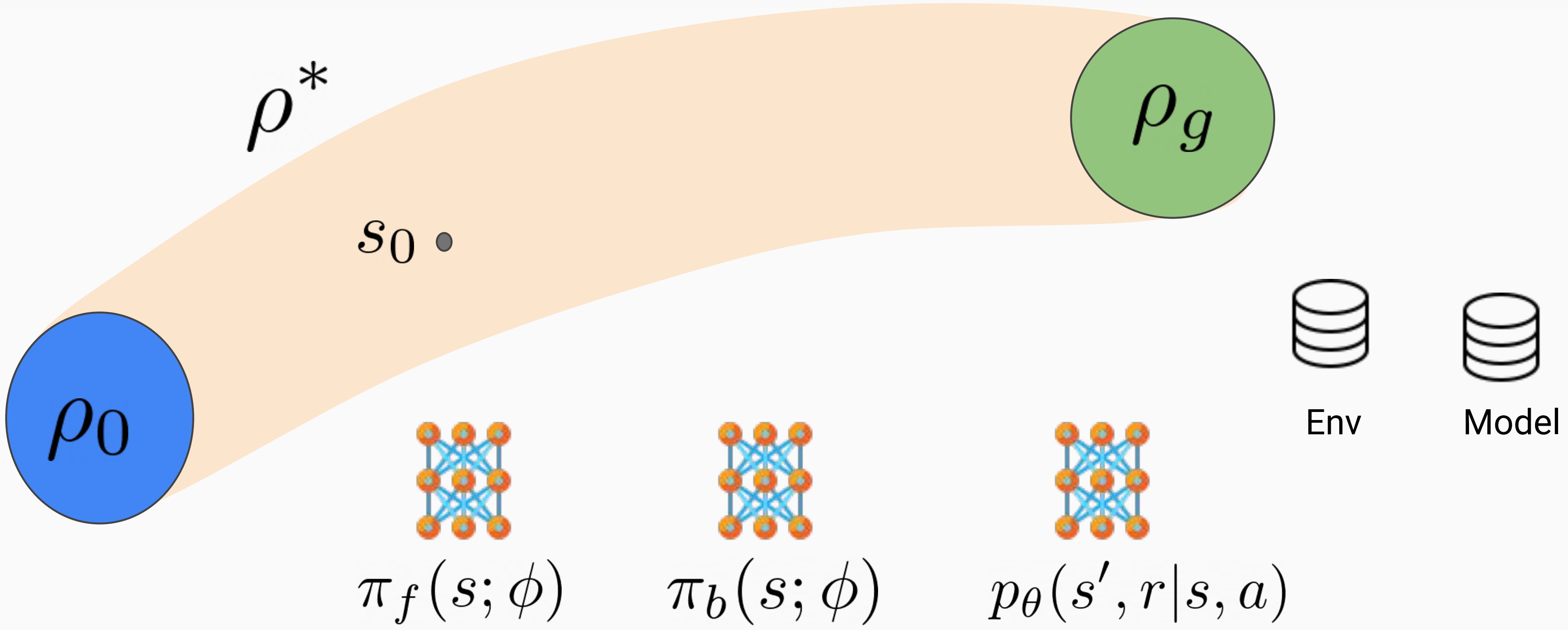
Approach: MBPO + MEDAL

- **Approach:** Leverage online dynamics and policy learning by hallucinating data with a global dynamics model $p(s', r|s, a; \omega)$, combing MBPO and MEDAL.

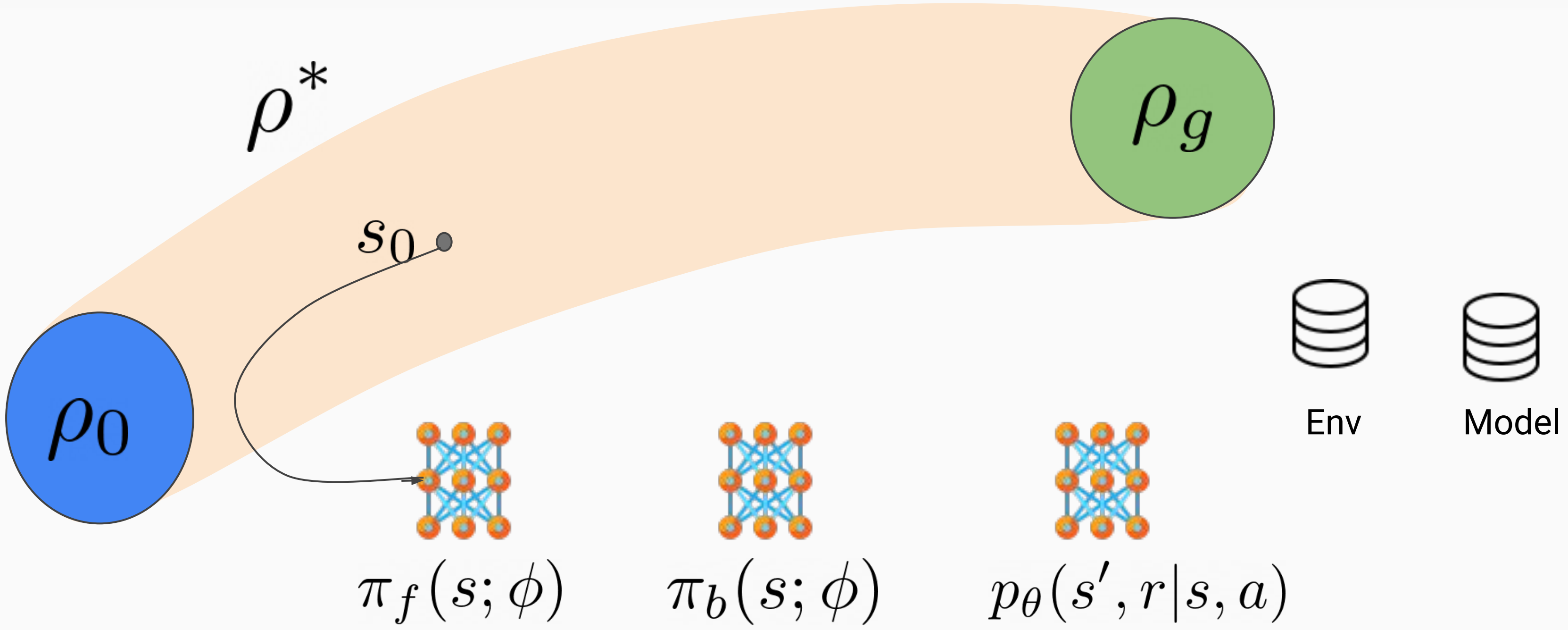
Approach: MBPO + MEDAL



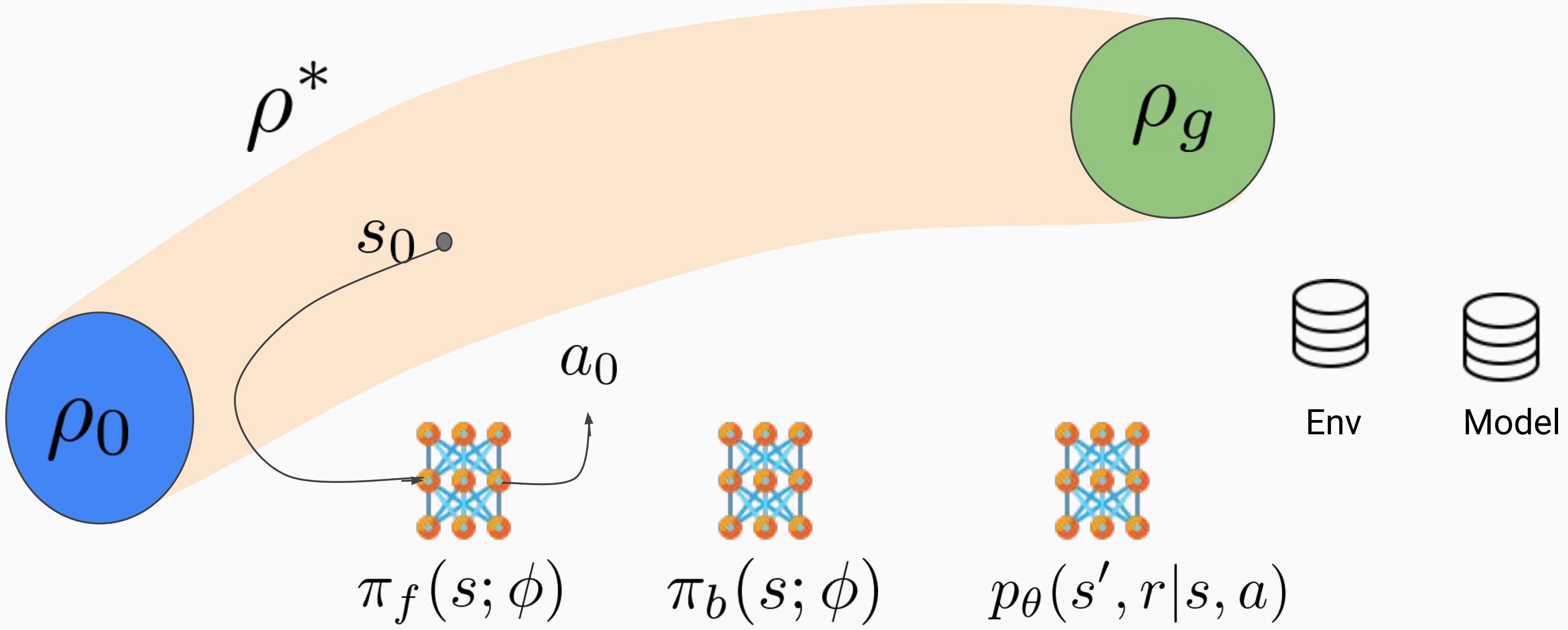
Approach: MBPO + MEDAL



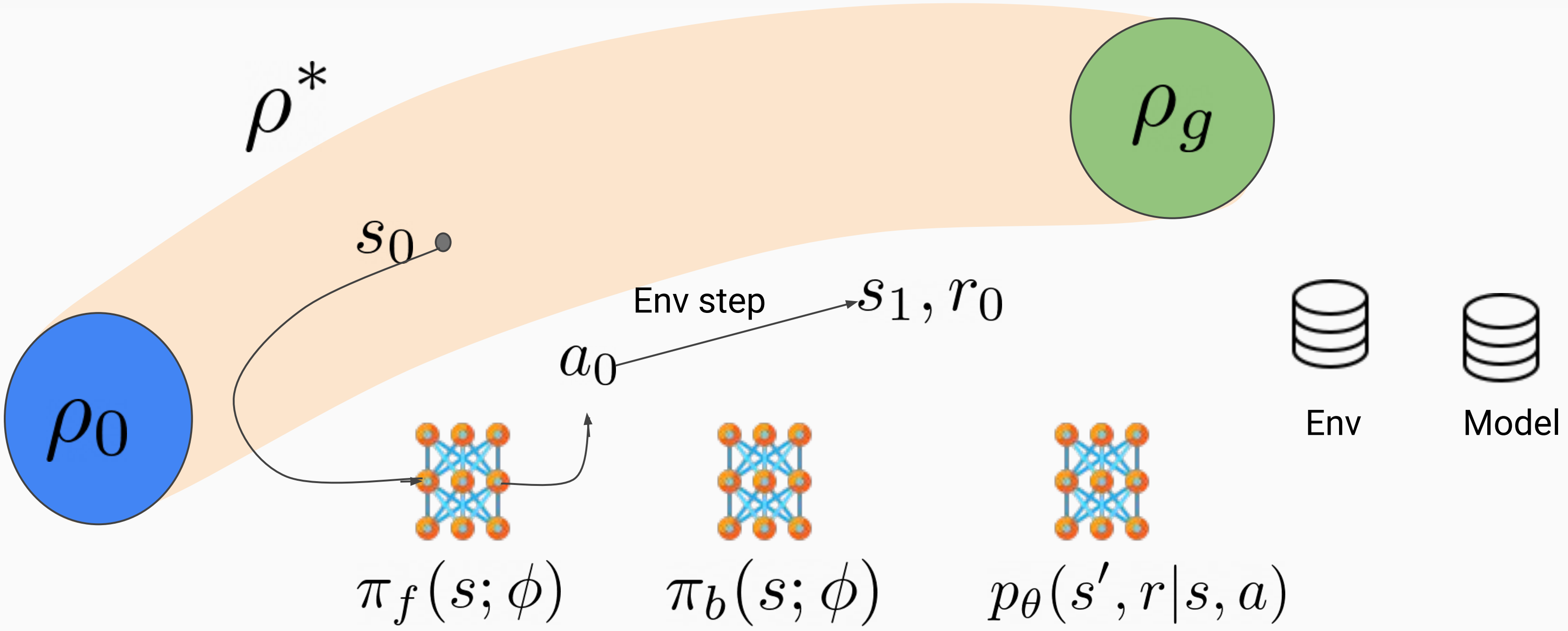
Approach: MBPO + MEDAL



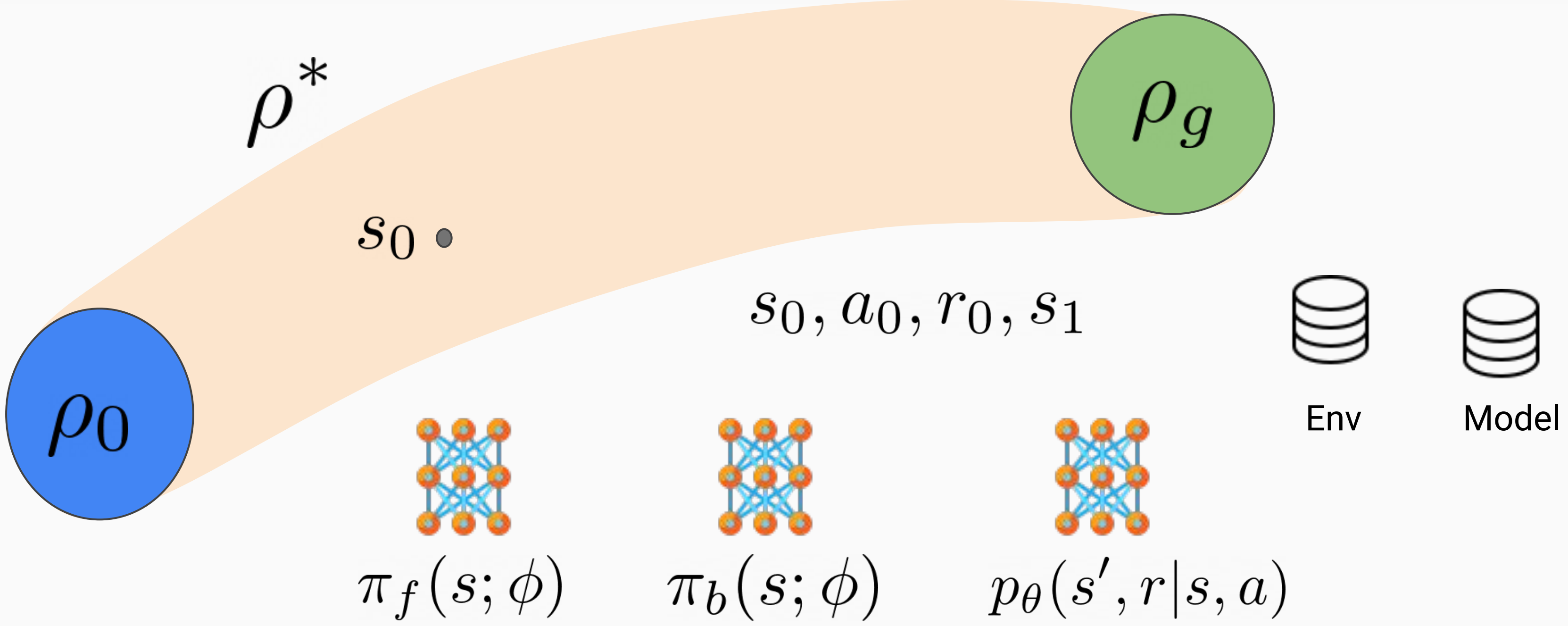
Approach: MBPO + MEDAL



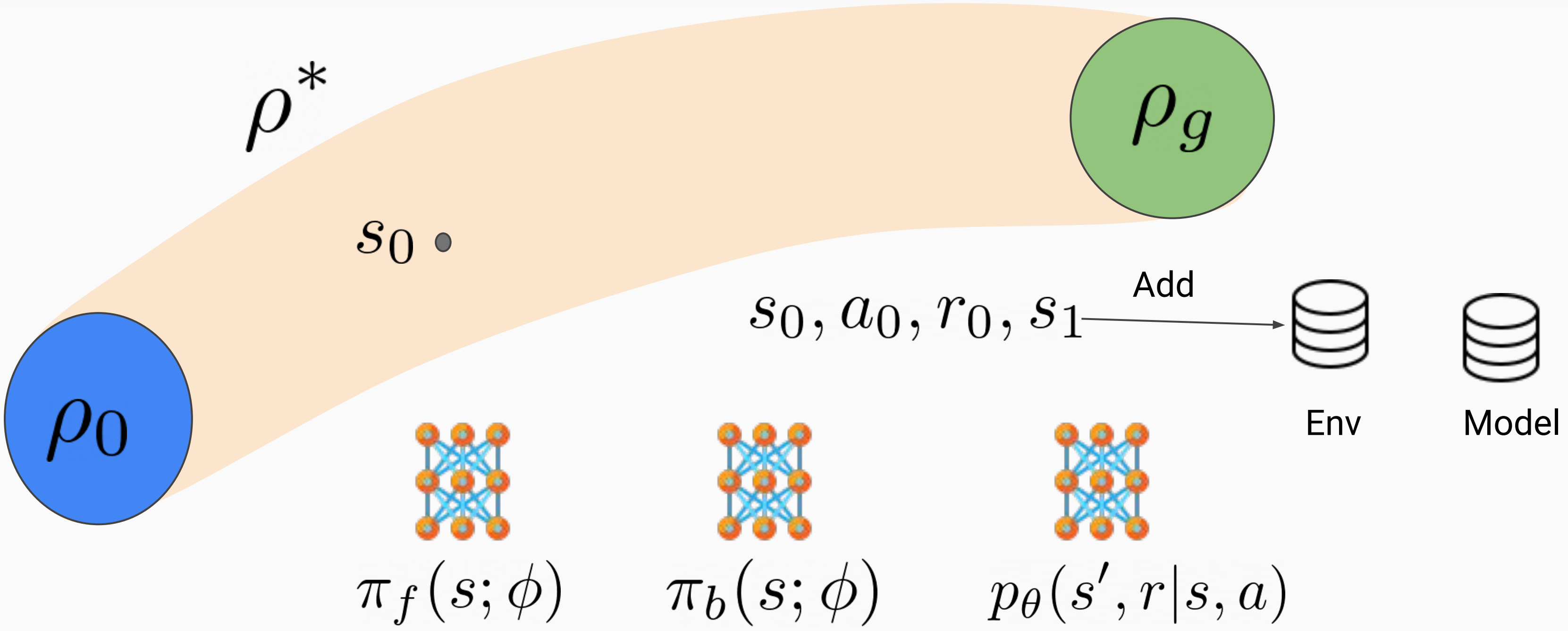
Approach: MBPO + MEDAL



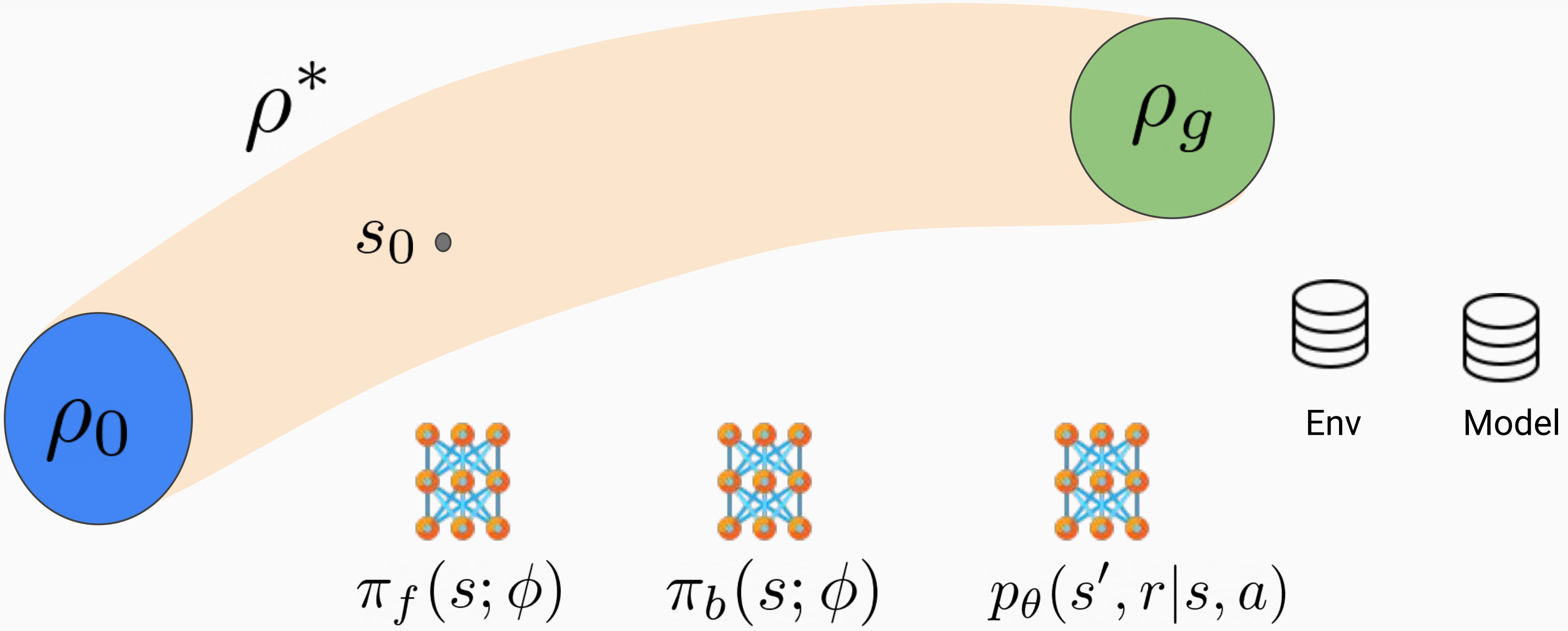
Approach: MBPO + MEDAL



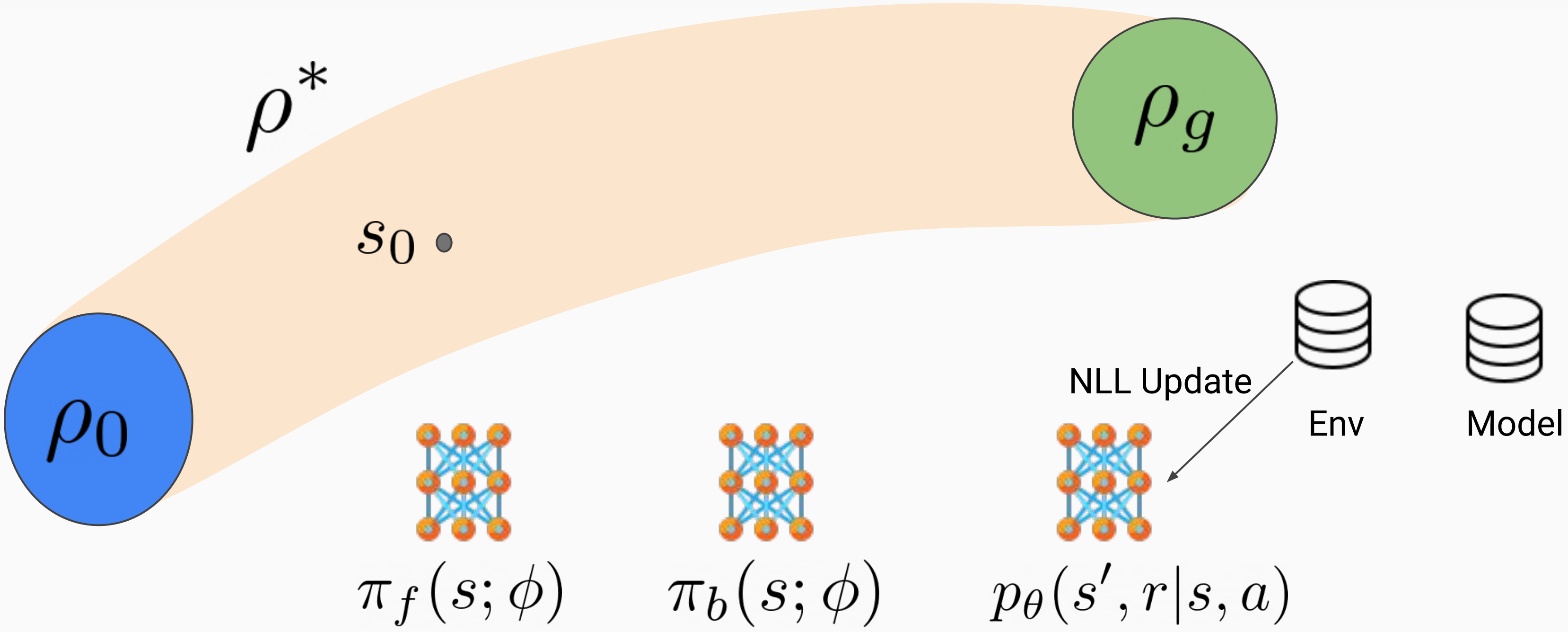
Approach: MBPO + MEDAL



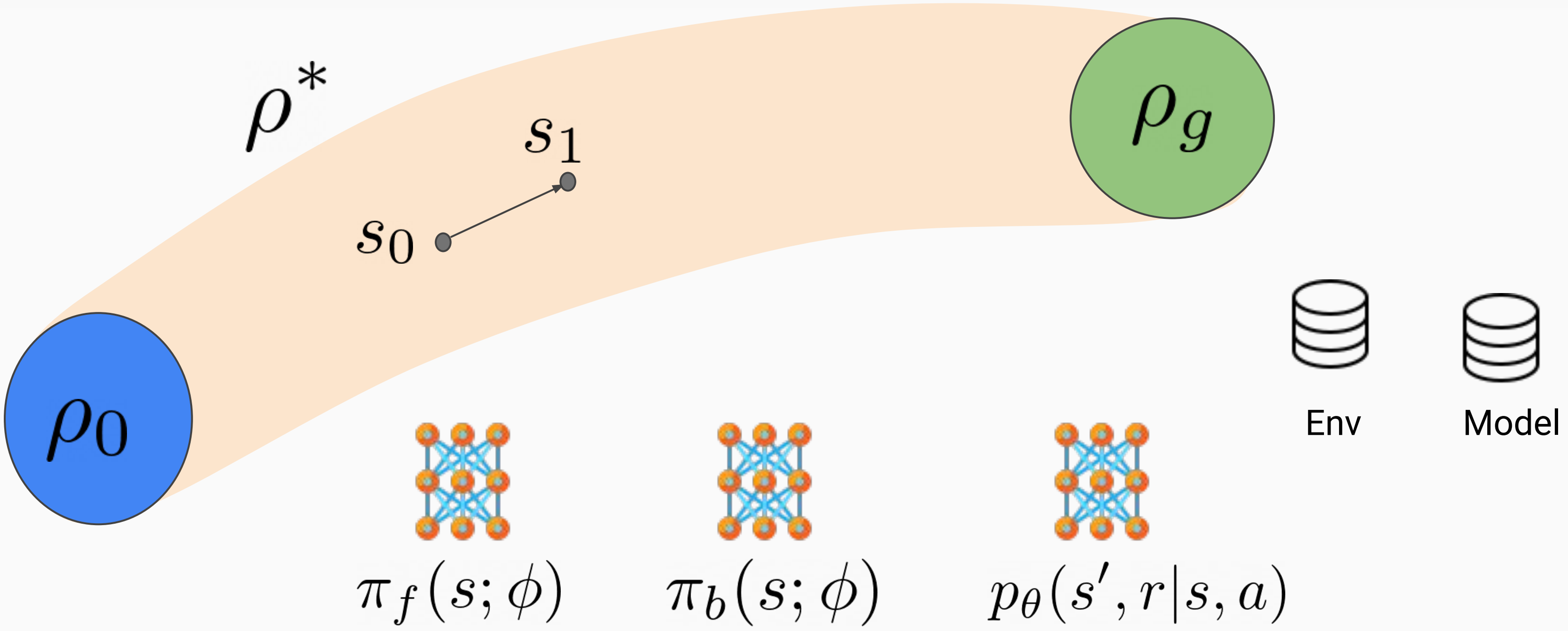
Approach: MBPO + MEDAL



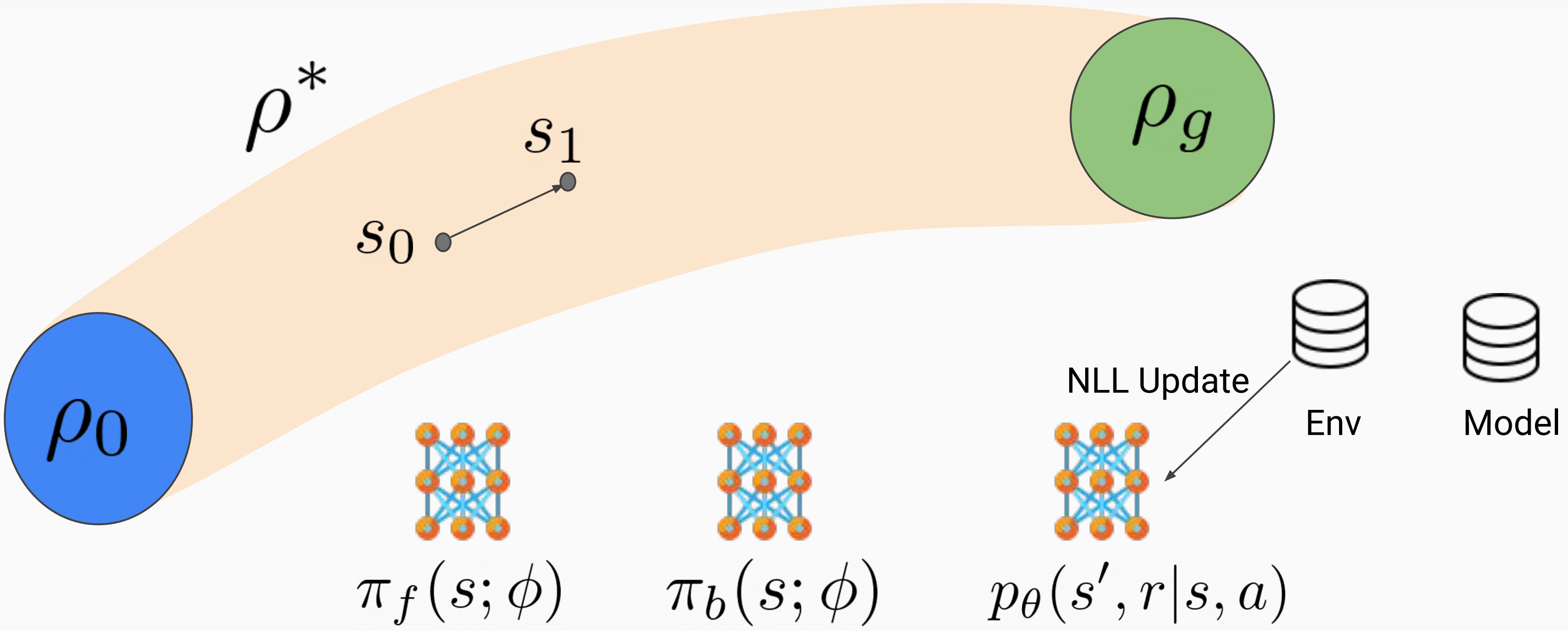
Approach: MBPO + MEDAL



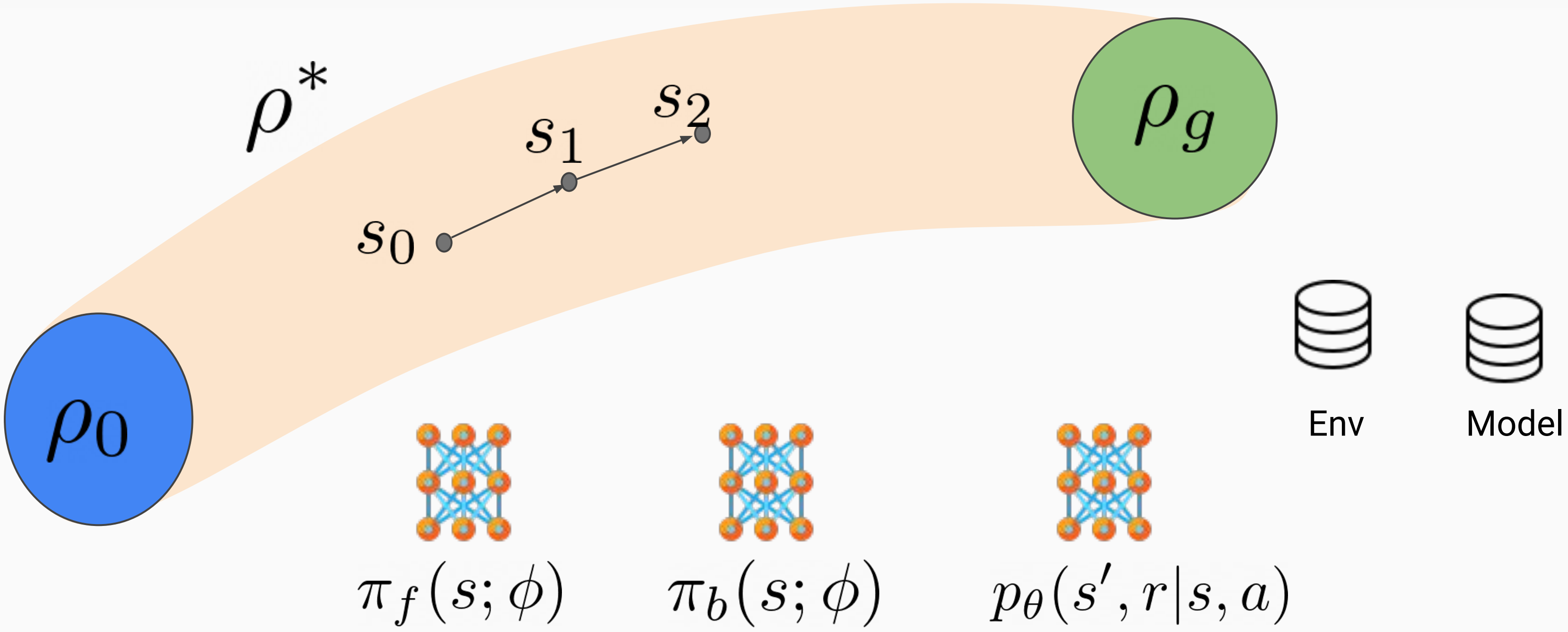
Approach: MBPO + MEDAL



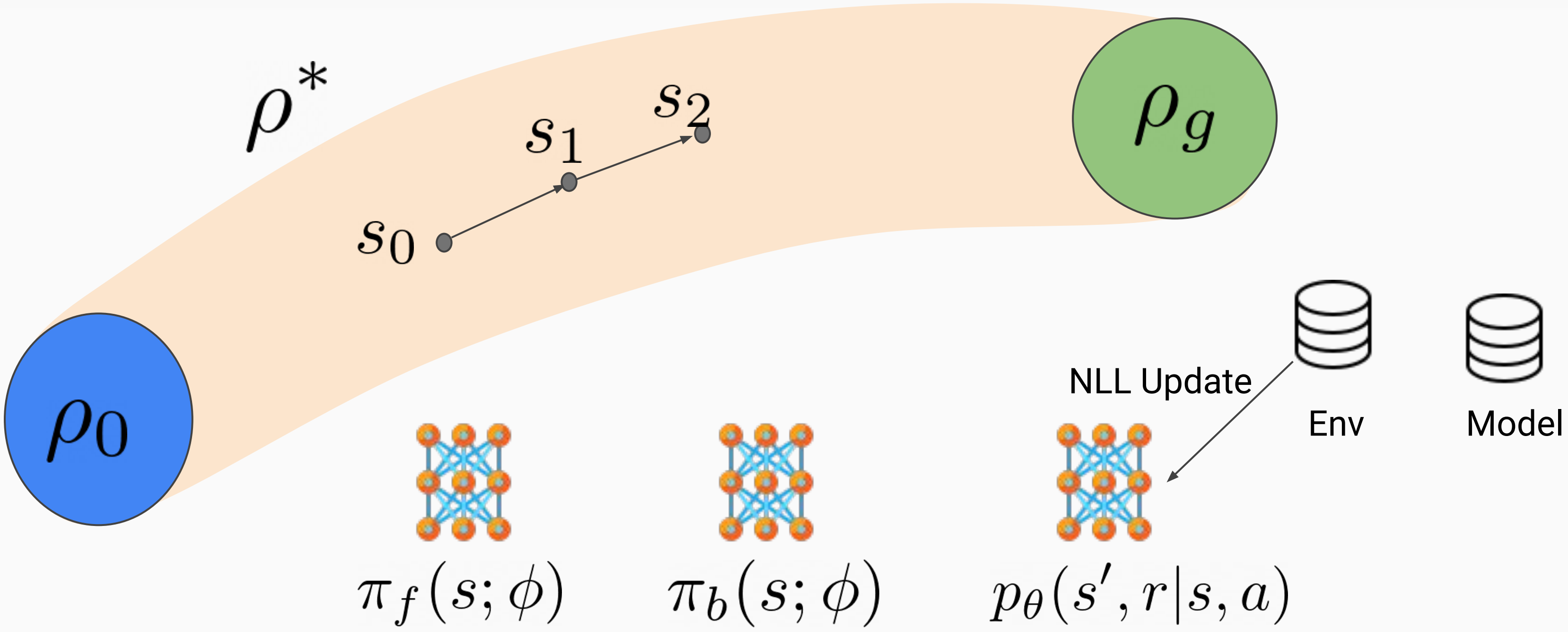
Approach: MBPO + MEDAL



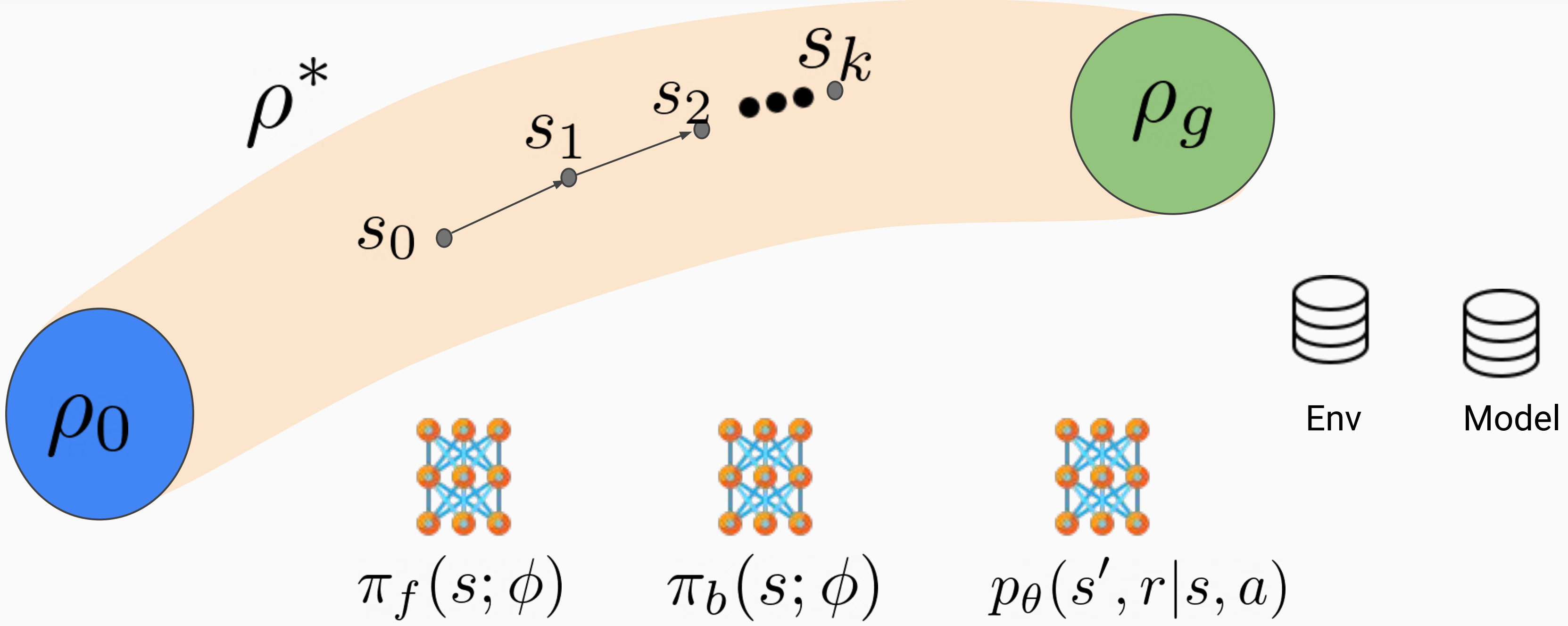
Approach: MBPO + MEDAL



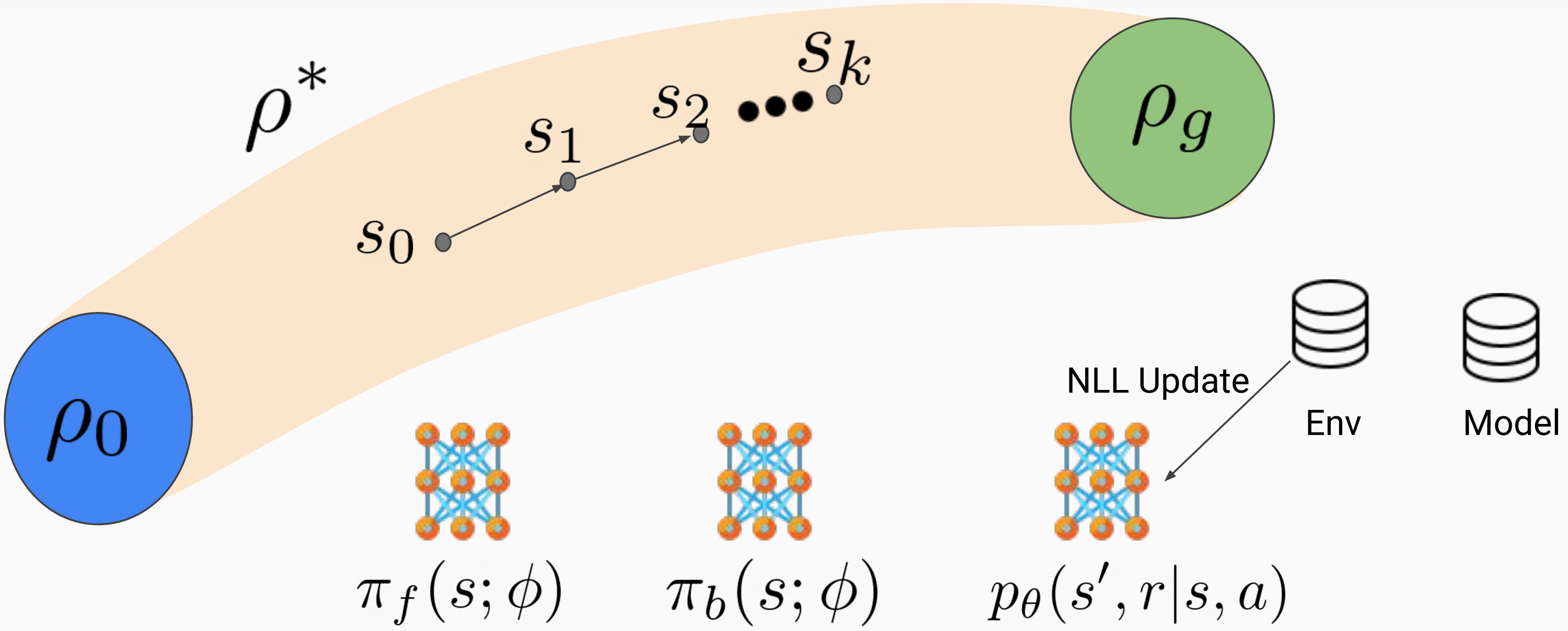
Approach: MBPO + MEDAL



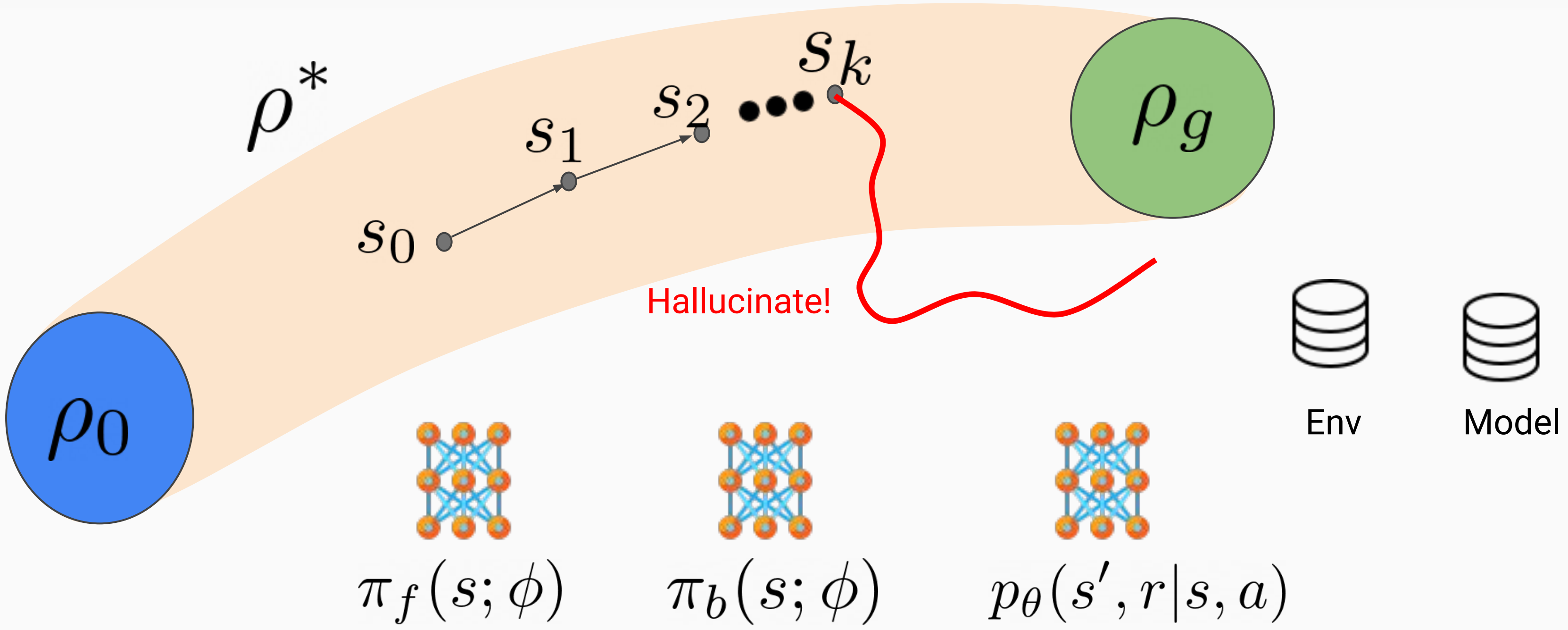
Approach: MBPO + MEDAL



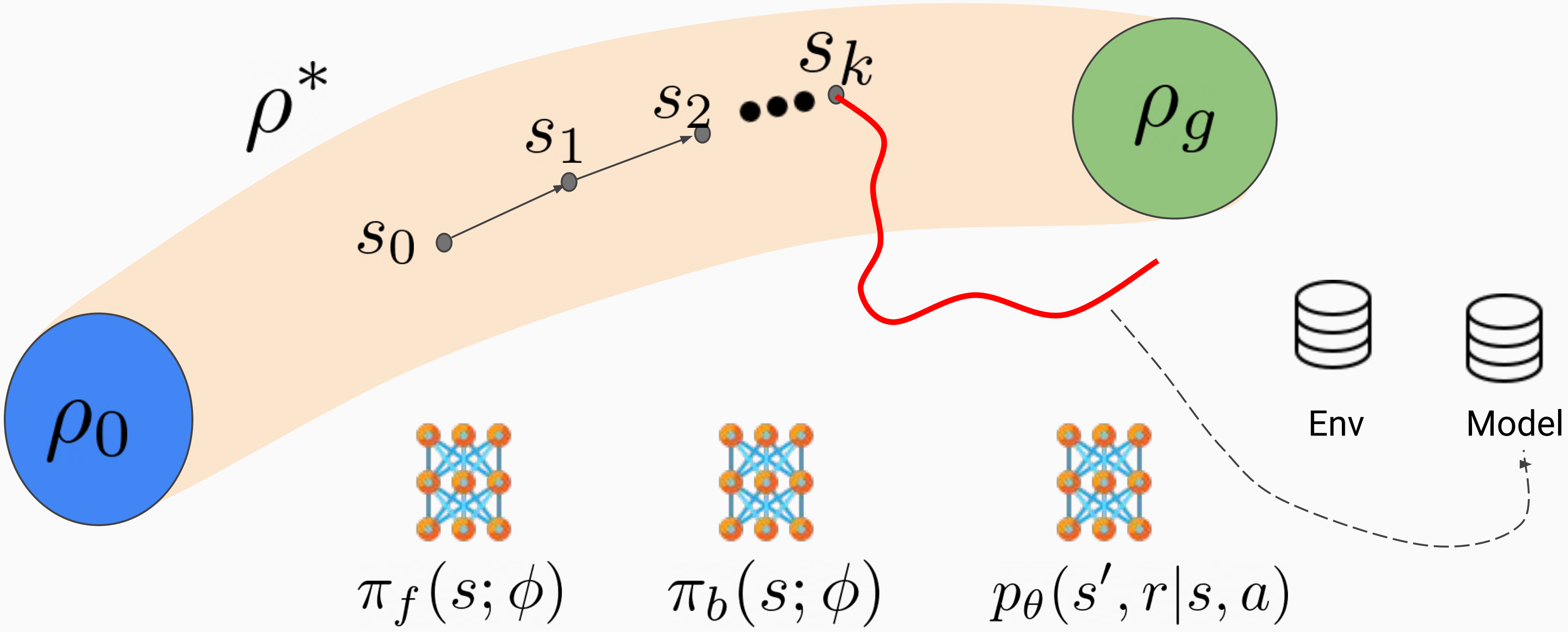
Approach: MBPO + MEDAL



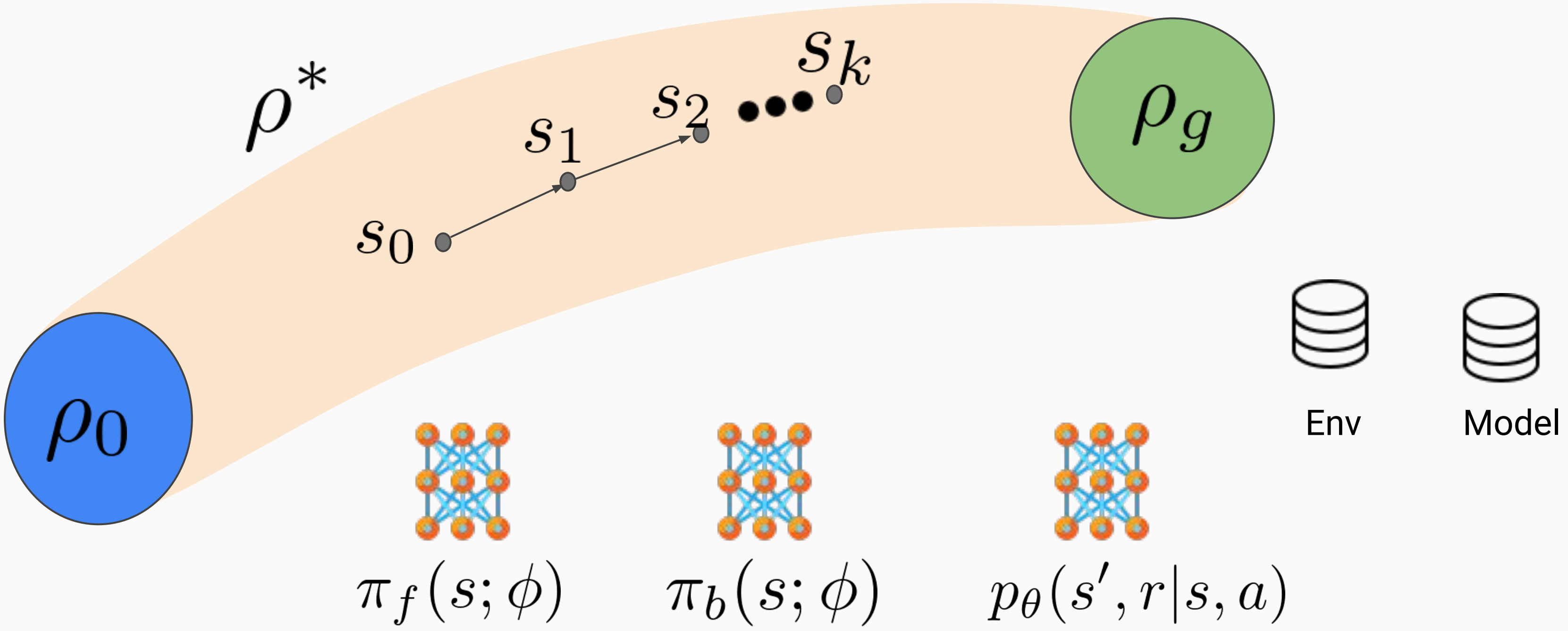
Approach: MBPO + MEDAL



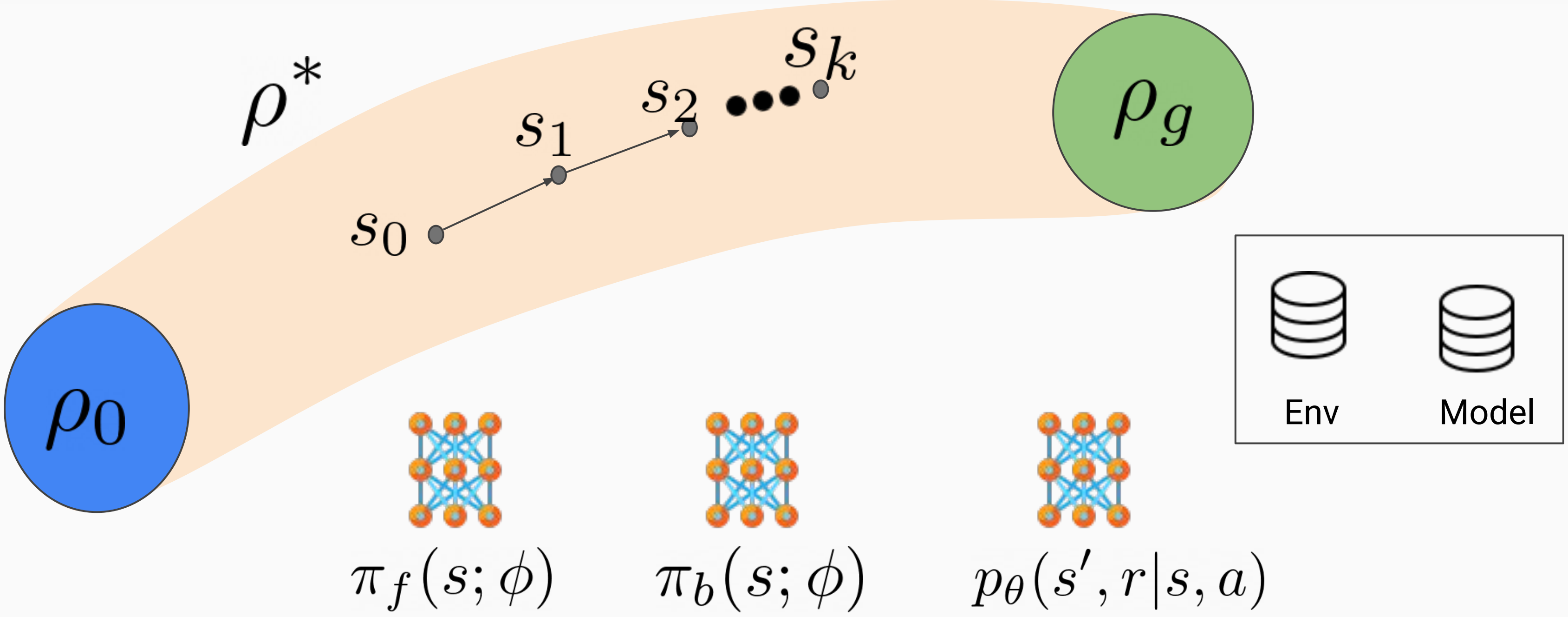
Approach: MBPO + MEDAL



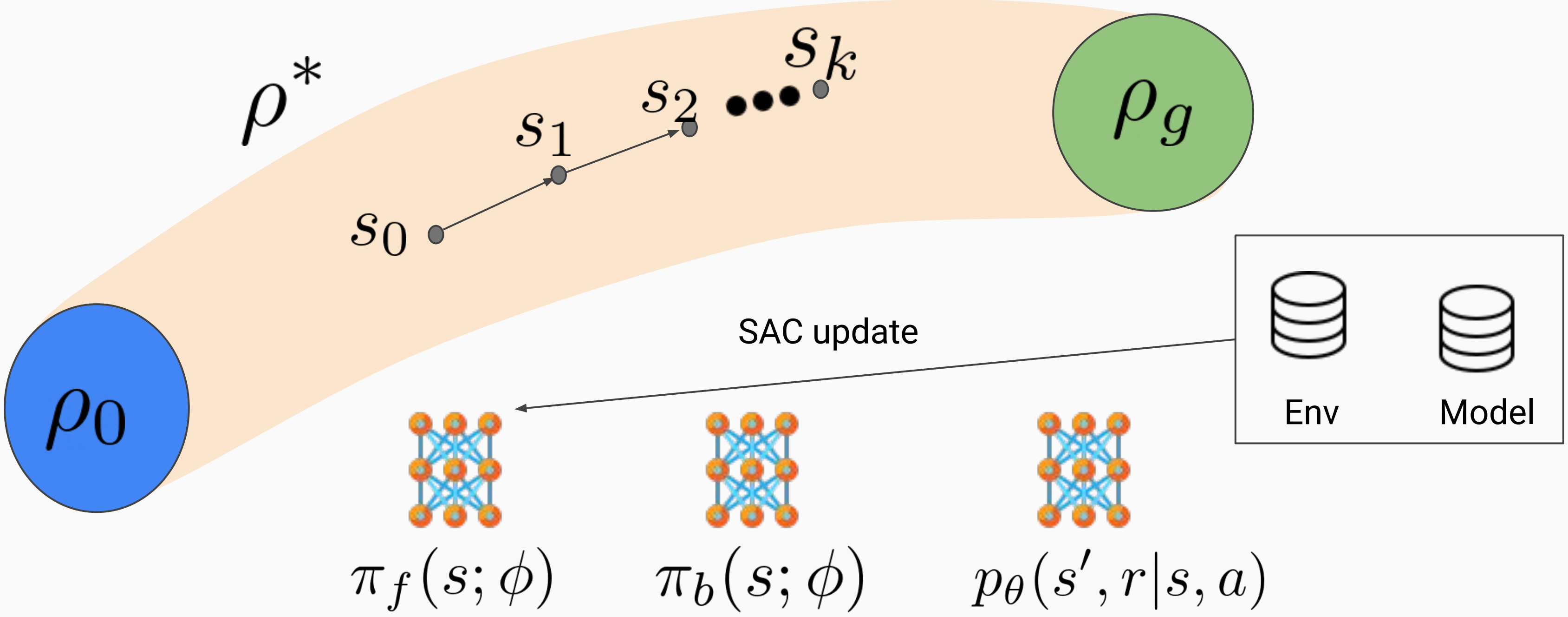
Approach: MBPO + MEDAL



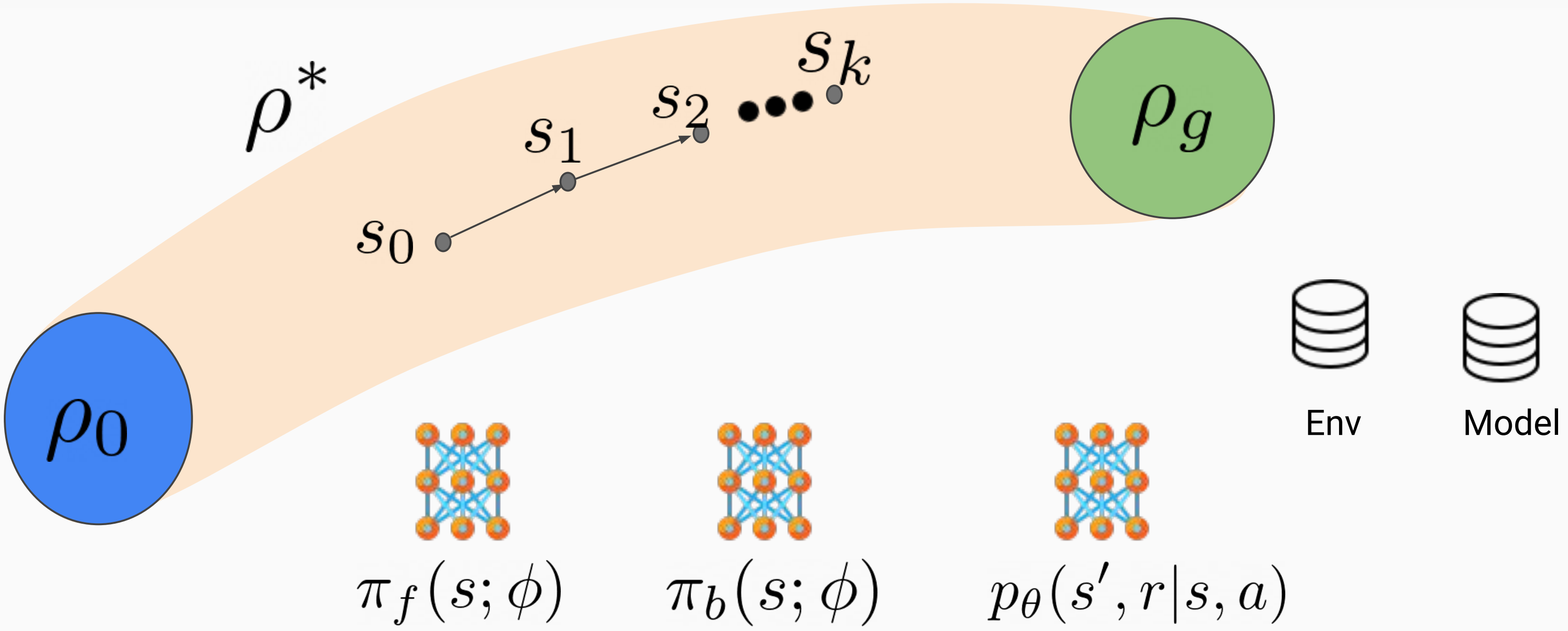
Approach: MBPO + MEDAL



Approach: MBPO + MEDAL



Approach: MBPO + MEDAL



Algorithm 1 Model-based Autonomous Reinforcement Learning (MBARL)

require: forward demos D_f
optional: backward demos D_b
initialize:
  $R_f, \pi_f(a|s;\phi_f), Q^{\pi_f}(s,a;\theta_f)$ ▷forward policy
  $R_b, \pi_f(a|s;\phi_b), Q^{\pi_b}(s,a;\theta_b)$ ▷backward policy
  $p(s', r|s,a;\omega) \sim \mathcal{N}(f^{\mu}_{\omega}, f^{\sigma}_{\omega})$ ▷Gaussian dynamics model
  $C(s;\psi)$ ▷state-space classifier
  $R_f \leftarrow R_f \cup D_f, R_b \leftarrow R_b \cup D_b$ ▷forward backward replay buffers
$s \sim \rho_0$ ▷sample initial state
while not done **do**
▷ Run forward policy for fixed number of steps until goal is reached, otherwise run backward policy
  if forward **then**
    $a \sim \pi_f(\cdot|s;\phi_f)$
    $s' \sim p(\cdot|s,a), r \leftarrow r(s,a)$
    $R_f \leftarrow R_f \cup \{(s,a,s',r)\}$
    update π_f, Q^{π_f}
    update p_ω with (s,a,s') via maximum likelihood
  else
    $a \sim \pi_b(\cdot|s;\phi_b)$
    $s' \sim p(\cdot|s,a), r \leftarrow -\log(1-C(s'))$
    $R_b \leftarrow R_b \cup \{(s,a,s',r)\}$
    update π_b, Q^{π_b}
    update p_ω with (s,a,s') via maximum likelihood

    ▷ Train discriminator every K steps
    if train-discriminator **then**
      sample positive states $S_p \sim D_f$
      sample negative states $S_n \sim D_b$
      update C on $S_p \cup S_n$
      ▷ Hallucination step using learned dynamics model
      for M model rollouts **do**
        for stage in (forward, backward) **do**
          policy $\pi = \pi_f$ if stage=forward else π_b
          replay buffer $R = R_f$ if stage=forward else R_b
          sample s_t uniformly from R
          perform k-step model rollout starting from s_t using policy π
          add sampled trajectory τ to R
        end for
      end for

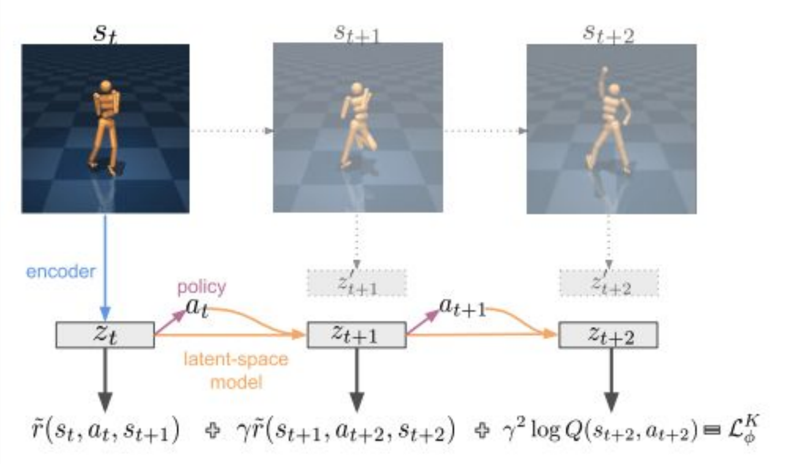
Objective Mismatch Problem

- Agent will seek states where dynamics model makes errors
- Discrepancy between policy objective and model objective is the **“objective mismatch problem”**
- Not good for high-dimensional state space

MBRL Objective

- Train dynamics model, representations, and policy to be self-consistent.
- Policy should only visit states where the model is accurate, the representation should encode information that is task-relevant and predictable.
- Learn model-based RL algorithm that automatically learns compact yet sufficient representations for model-based reasoning.

Aligned Latent Models (ALM)



MBRL algorithm that jointly optimizes the observation representations, a model that predicts those representations, and a policy that acts based on those representations.

Preliminaries

- Markov Decision Process: $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, r, \rho_0)$
- Learn policy that maximizes the discounted sum of expected rewards within an infinite-horizon episode:

$$\max_{\pi} \mathbb{E}_{s_{t+1} \sim p(\cdot|s_t, a_t), a_t \sim \pi(\cdot|s_t)} \left[(1-\gamma) \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right]$$

Components of ALM

- Encoder: $e_\phi(z_t \mid s_t)$
- Dynamics model of representations: $m_\phi(z_{t+1} \mid z_t, a_t)$
- Policy: $\pi_\phi(a_t \mid z_t)$

- RL objective: $\mathbb{E}_{p(\tau)}[R(\tau)]$
- Write distribution over trajectories as:

$$p_\phi(\tau) \triangleq p_0(s_0) \prod_{t=0}^{\infty} p(s_{t+1} \mid s_t, a_t) \pi_\phi(a_t \mid z_t) e_\phi(z_t \mid s_t)$$

Evidence lower bound (from Jensen's inequality):

$$\log \mathbb{E}_{p(\tau)}[R(\tau)] \geq \mathbb{E}_{q(\tau)}\left[\log R(\tau) + \log p(\tau) - \log q(\tau)\right]$$

Only sample compact representations:

$$q_\phi^K(\tau) = p_0(s_0) e_\phi(z_0 \mid s_0) \pi_\phi(a_0 \mid z_0) \prod_{t=1}^{K} p(s_t \mid s_{t-1}, a_{t-1}) m_\phi(z_t \mid z_{t-1}, a_{t-1}) \pi_\phi(a_t \mid z_t)$$

$$\cdot \prod_{t=K+1}^{\infty} p(s_t \mid s_{t-1}, a_{t-1}) m_\phi(z_t \mid z_{t-1}, a_{t-1}) \pi_\phi(a_t \mid z_t).$$

ALM Objective

Evidence lower bound (ELBO):

$$\mathcal{L}_\phi^K \triangleq \mathbb{E}_{q_\phi^K(\tau)}\left[\left(\sum_{t=0}^{K-1}\gamma^t \tilde{r}(s_t, a_t, s_{t+1})\right) + \gamma^K \log Q(s_K, a_K)\right],$$

$$\text{where } \tilde{r}(s_t, a_t, s_{t+1}) = \underbrace{(1-\gamma)\log r(s_t, a_t)}_{(a)} + \underbrace{\log e_\phi(z_{t+1} \mid s_{t+1}) - \log m_\phi(z_{t+1} \mid z_t, a_t)}_{(b)}$$

Autonomous RL using Aligned Latent Models

Algorithm 1 Model-based Autonomous Reinforcement Learning using Aligned Latent models (MBARL)

require: forward demos D_f
optional: backward demos D_b
initialize:
$e_\phi(z|s)$ ▷encoder
$m_\phi(z'|z,a)$ ▷latent model
$\pi^f_\phi(a|s), \pi^b_\phi(a|s)$ ▷forward/backward policy
$C_\theta(z',a,z)$ ▷classifier to distinguish $e_\phi and m_\phi$
$f_\theta(s)$ ▷state-space discriminator
$r^f_\theta(z,a), r^b_\theta(z,a)$ ▷forward/backward reward
$Q^f_\theta(z,a), Q^b_\theta(z,a)$ ▷Q-functions
R^f, R^b ▷forward/backward replay buffer

$s \sim \rho_0$ ▷sample initial state
while not done **do**
▷ Run forward policy for fixed number of steps until goal is reached, otherwise run backward policy
if forward **then**
Select action $a = \pi^f_\phi(\cdot|e_\phi(s)) + \mathcal{N}$
$s' \sim p(\cdot|s,a), r \leftarrow r(s,a)$
$R_f \leftarrow R_f \cup \{(s,a,s',r)\}$
Sample length-K sequence $\{s_i, a_i, s_{i+1}\}_{i=t}^{t+K-1} \sim R_f$
Compute lower bound using off-policy actions

$$\begin{aligned}\mathcal{L}^K_{e_\phi,m_\phi}(\{s_i,a_i,s_{i+1}\}_{i=t}^{t+K}) = E_{\substack{e_\phi(z_{i=t}|s_t)\\ m_\phi(z_{i>t}|z_{t:i-1},a_{i-1})}} \Bigg[\gamma^K Q^f_\theta(z_K, \pi^f(z_K)) \\ + \sum_{i=t}^{t+K-1} \gamma^i (r^f_\theta(z_i,a_i) - KL(e_{\phi_{\text{targ}}}(z_{i+1}|s_{i+1}) || m_\phi(z_{i+1}|z_i,a_i)\Bigg]\end{aligned} \tag{1}$$

Update encoder e_ϕ and latent model m_ϕ by gradient ascent on off-policy lower bound $\mathcal{L}^K_{e_\phi,m_\phi}$
Compute lower bound using on-policy actions

$$\mathcal{L}^K_{\pi^f_\phi}(\{s_t\}) = E_{q_\phi(z_{t:K},a_{t:K}|s_t)}\left[\sum_{i=t}^{t+K-1} \gamma^i \left(r^f_\theta(z_i,a_i) + c\log\frac{C_\theta(z_{i+1},a_i,z_i)}{1-C_\theta(z_{i+1},a_i,z_i)}\right) + \gamma^K Q^f_\theta(z_K,\pi^f(z_K))\right]$$

Update policy by gradient ascent on on-policy lower bound: $\mathcal{L}^K_{\pi^f_\phi}$
Update latent classifier C_θ, Q-function Q^f_θ and r^f_θ via gradient ascent on: $\mathcal{L}_{C_\theta}, \mathcal{L}_{Q^f_\theta}, \mathcal{L}_{r^f_\theta}$

else

Select action $a = \pi^b_\phi(\cdot|e_\phi(s)) + \mathcal{N}$

$s' \sim p(\cdot|s,a), r \leftarrow -\log(1 - f_\theta(e_\phi(s')))$ ▷Distribution matching ρ^* and ρ^b

$R_b \leftarrow R_b \cup \{(s,a,s',r)\}$

Sample length-K sequence $\{s_i, a_i, s_{i+1}\}_{i=t}^{t+K-1} \sim R_f$

Compute lower bound using off-policy actions

$$\mathcal{L}^K_{e_\phi,m_\phi}(\{s_i,a_i,s_{i+1}\}_{i=t}^{t+K}) = E_{\substack{e_\phi(z_{i=t}|s_t) \\ m_\phi(z_{i>t}|z_{t:i-1},a_{i-1})}}\Bigg[\gamma^K Q^b_\theta(z_K, \pi^b(z_K)) + \sum_{i=t}^{t+K-1} \gamma^i(r^b_\theta(z_i,a_i) - KL(e_{\phi_{\text{targ}}}(z_{i+1}|s_{i+1})||m_\phi(z_{i+1}|z_i,a_i)\Bigg] \tag{2}$$

Update encoder e_ϕ and latent model m_ϕ by gradient ascent on off-policy lower bound $\mathcal{L}^K_{e_\phi,m_\phi}$

Compute lower bound using on-policy actions

$$\mathcal{L}^K_{\pi^b_\phi}(\{s_t\}) = E_{q_\phi(z_{t:K},a_{t:K}|s_t)}\left[\sum_{i=t}^{t+K-1}\gamma^i\left(r^b_\theta(z_i,a_i) + c\log\frac{C_\theta(z_{i+1},a_i,z_i)}{1 - C_\theta(z_{i+1},a_i,z_i)}\right) + \gamma^K Q^b_\theta(z_K,\pi^b(z_K))\right]$$

Update policy by gradient ascent on on-policy lower bound: $\mathcal{L}^K_{\pi^b_\phi}$

Update latent classifier C_θ, Q-function Q^b_θ and r^b_θ via gradient ascent on: $\mathcal{L}_{C_\theta}, \mathcal{L}_{Q^f_\theta}, \mathcal{L}_{r^f_\theta}$

end if

▷ Train discriminator every K steps

if train-discriminator **then**

sample positive states $S_p \sim D_f$

sample negative states $S_n \sim D_b$

update f_θ on $S_p \cup S_n$

s← s'

Next Steps

Sketch of Coupled Algorithm

- **Exploration:** Explore and collect data from the environment s.t. we learn a "good" model $p_\theta(s', r|s, a)$
- **Evaluation:** agent learns a good policy π from dynamics model

Exploration Policy: Maximum Entropy

- Exploration policy π_{exp} matches the demonstration state distribution:

$$D_{\text{KL}}(\rho^{\pi_{\text{exp}}}(s)||\rho^*(s))$$

- with high entropy for *good coverage*:

$$\begin{aligned} \text{maximize } & H(\rho^{\pi_{\text{exp}}}(s)) \\ \text{subject to } & D_{\text{KL}}(\rho^{\pi_{\text{exp}}}(s)||\rho^*(s)) < \varepsilon \end{aligned}$$

Thanks to Archit & Chelsea!