Distillation of RL Policies through Bisimilar Latent Models with Formal Guarantees

Florent Delgrange, Ann Nowé, Guillermo A. Pérez

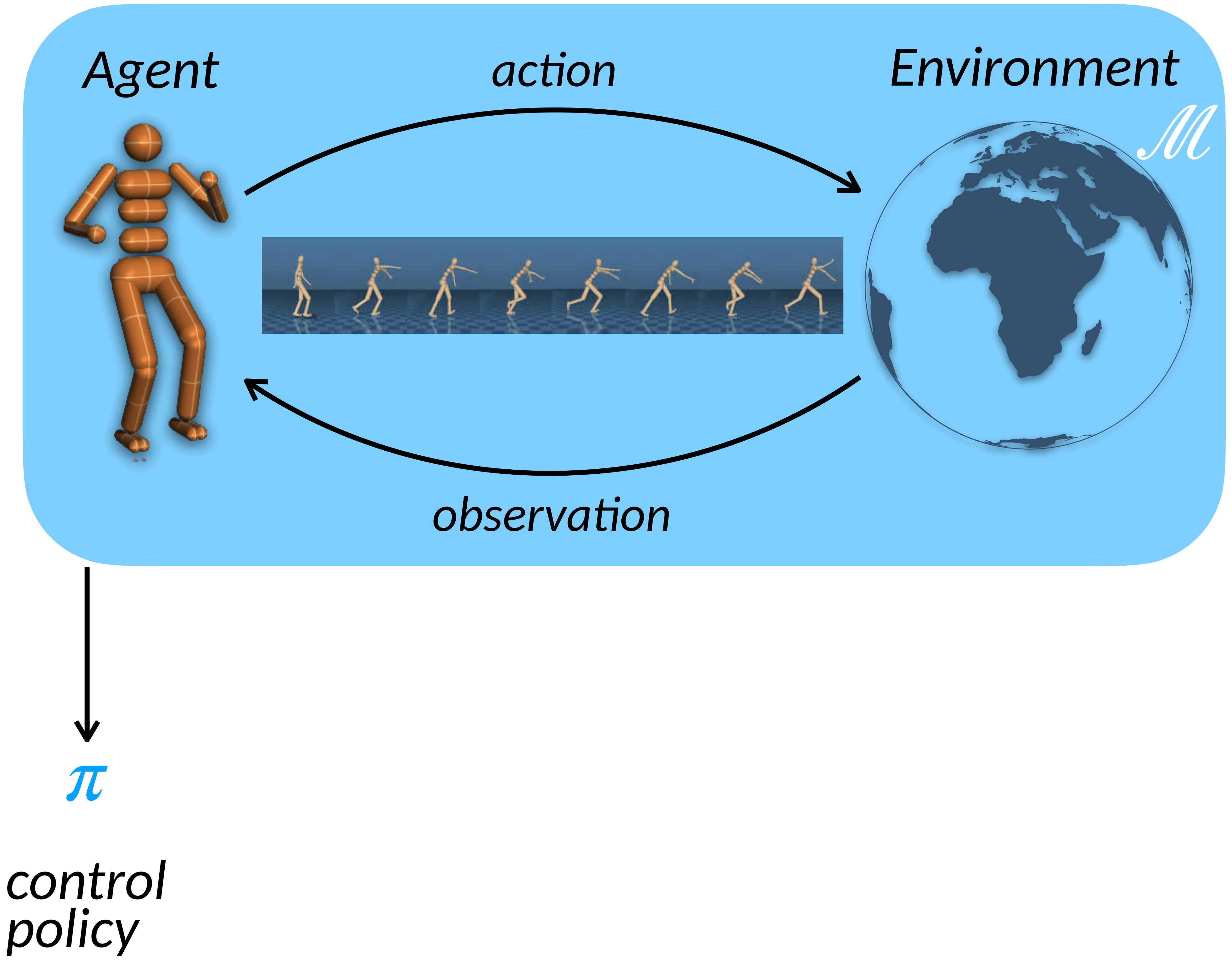
VUB

ARTIFICIAL INTELLIGENCE RESEARCH GROUP

Universiteit Antwerpen

Overview

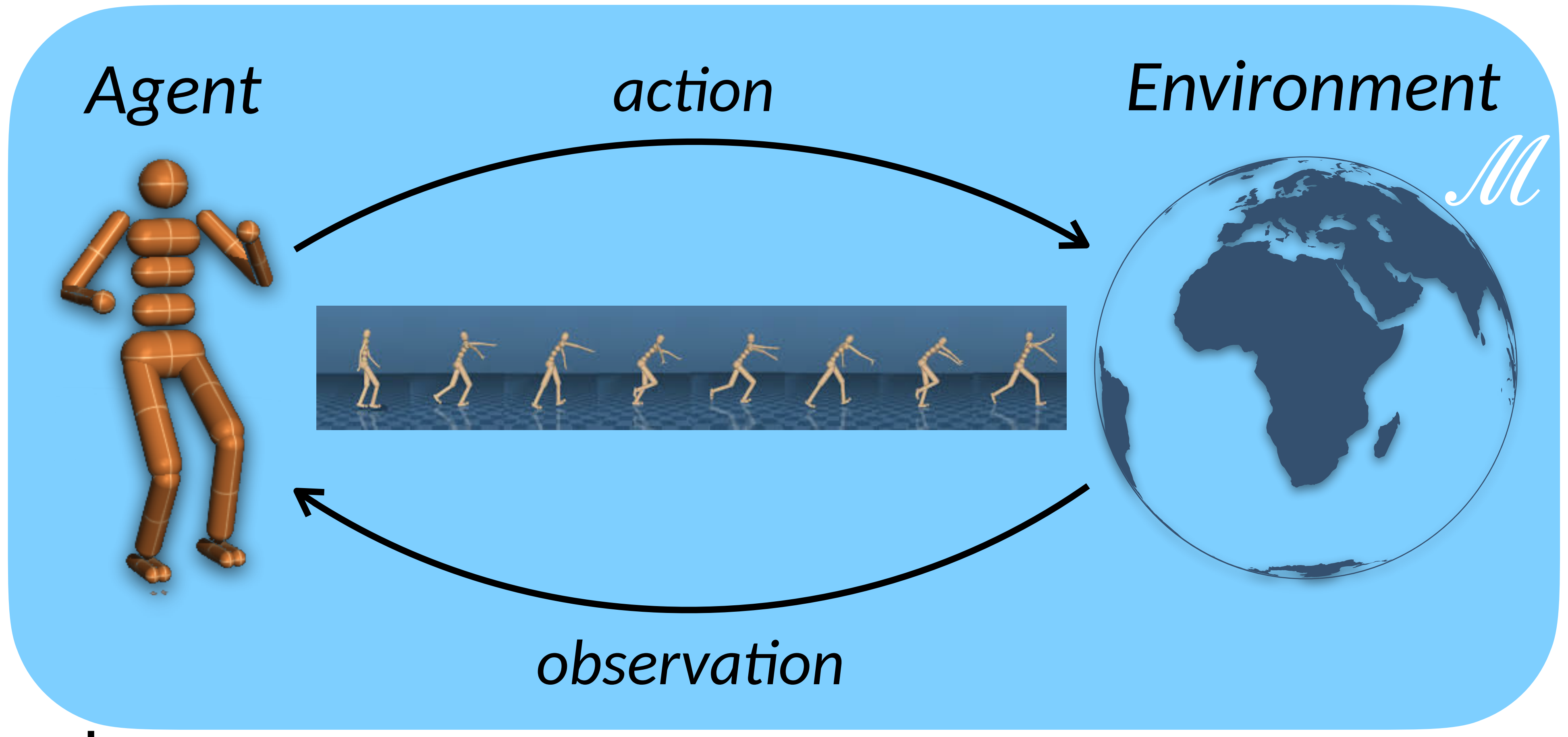
Reinforcement Learning



- Unknown environment
- Continuous state/action spaces

Overview

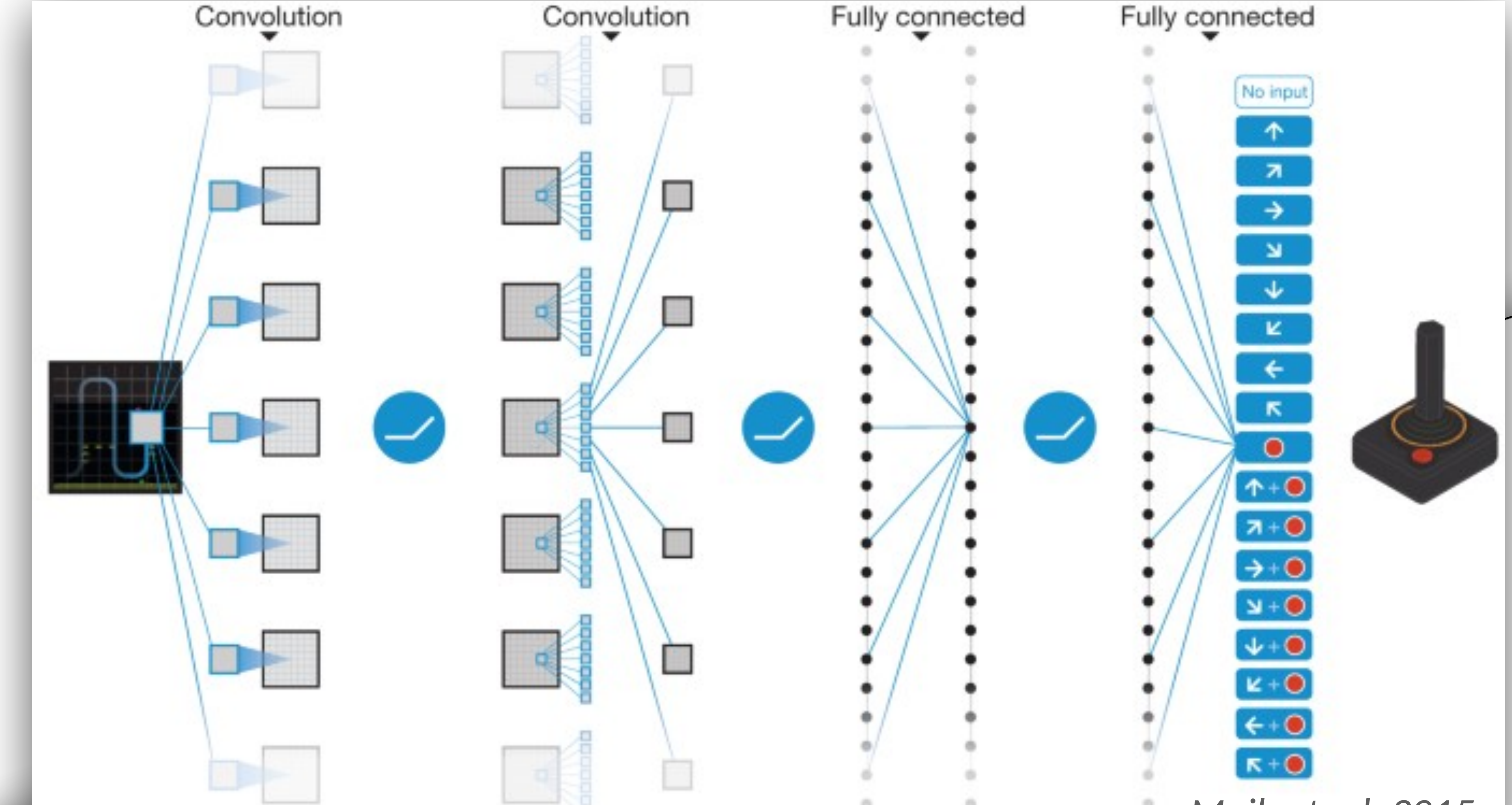
Reinforcement Learning



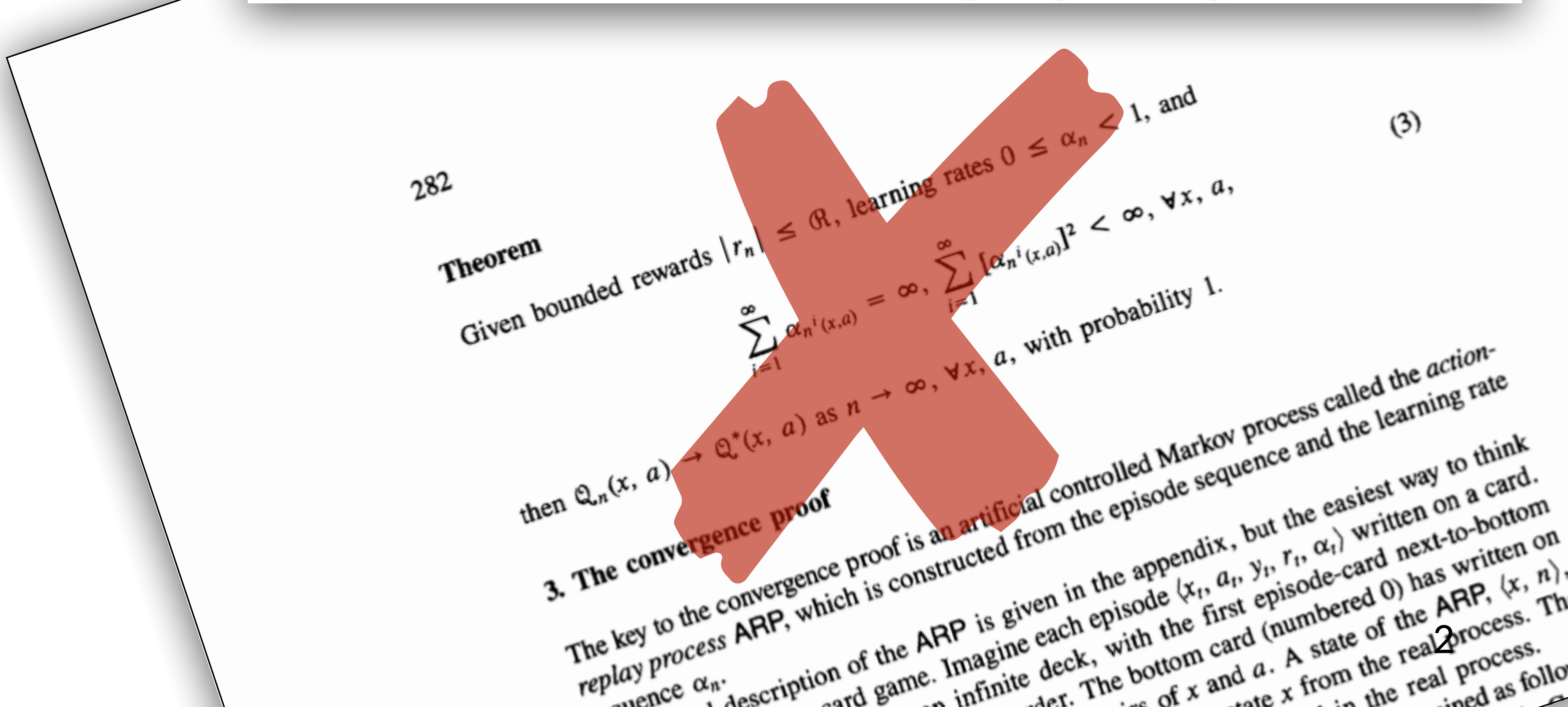
π

control policy

- Unknown environment
- Continuous state/action spaces



Mnih et. al, 2015



282

Theorem

Given bounded rewards $|r_n| \le \mathcal{R}$, learning rates $0 \le \alpha_n < 1$, and

$$\sum_{i=1}^{\infty} \alpha_{n^i(x,a)} = \infty, \quad \sum_{i=1}^{\infty} [\alpha_{n^i(x,a)}]^2 < \infty, \quad \forall x, a, \qquad (3)$$

then $Q_n(x, a) \to Q^*(x, a)$ as $n \to \infty$, $\forall x, a$, with probability 1.

3. The convergence proof

The key to the convergence proof is an artificial controlled Markov process called the *action-replay process* ARP, which is constructed from the episode sequence and the learning rate sequence α_n.

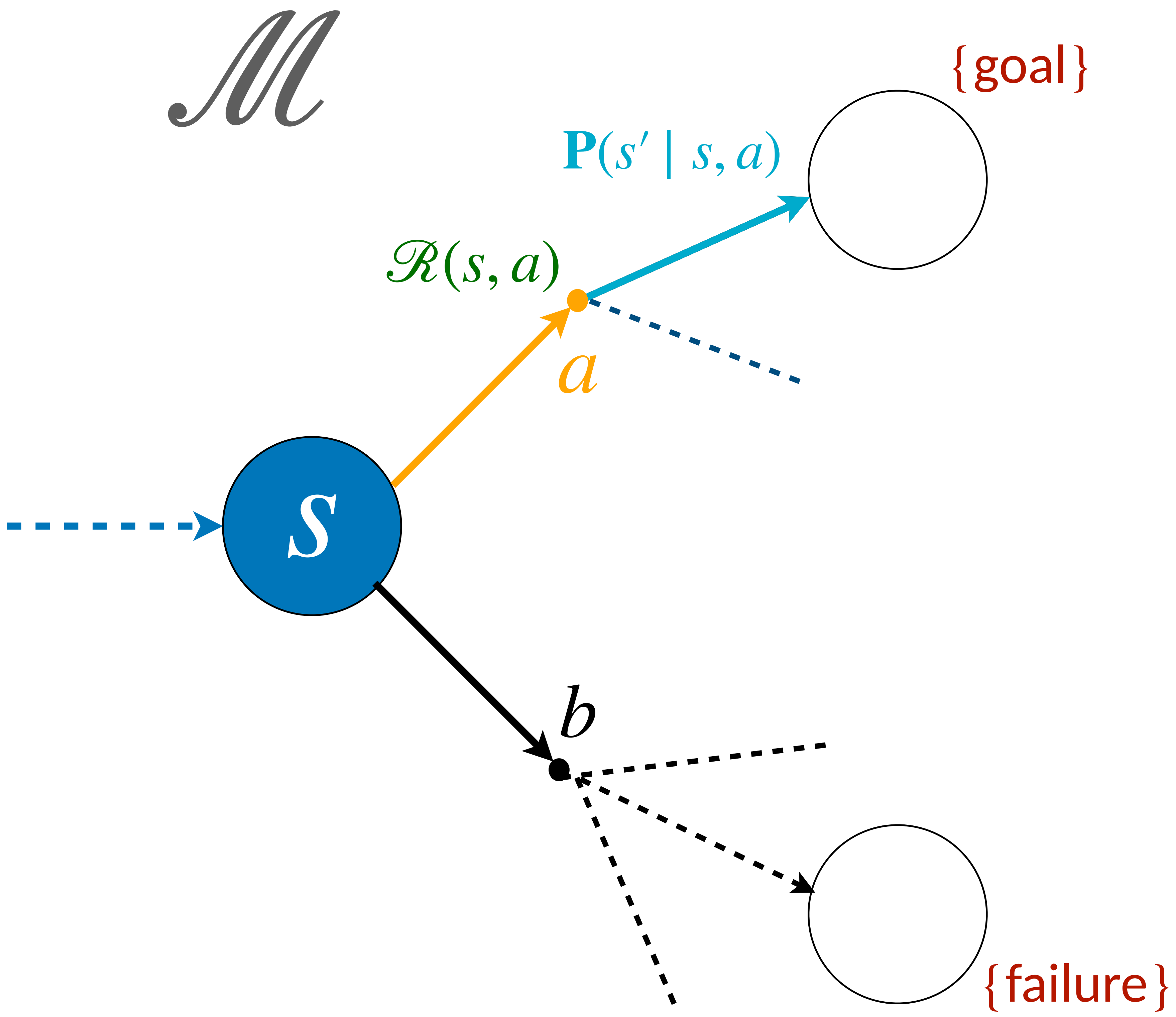
Overview

Reinforcement Learning Policies with Formal Guarantees

- Unknown environment
- Continuous state/action spaces

- Full knowledge of the model of the interaction
- Exhaustive exploration of the model
- Sensitive to the state space explosion problem

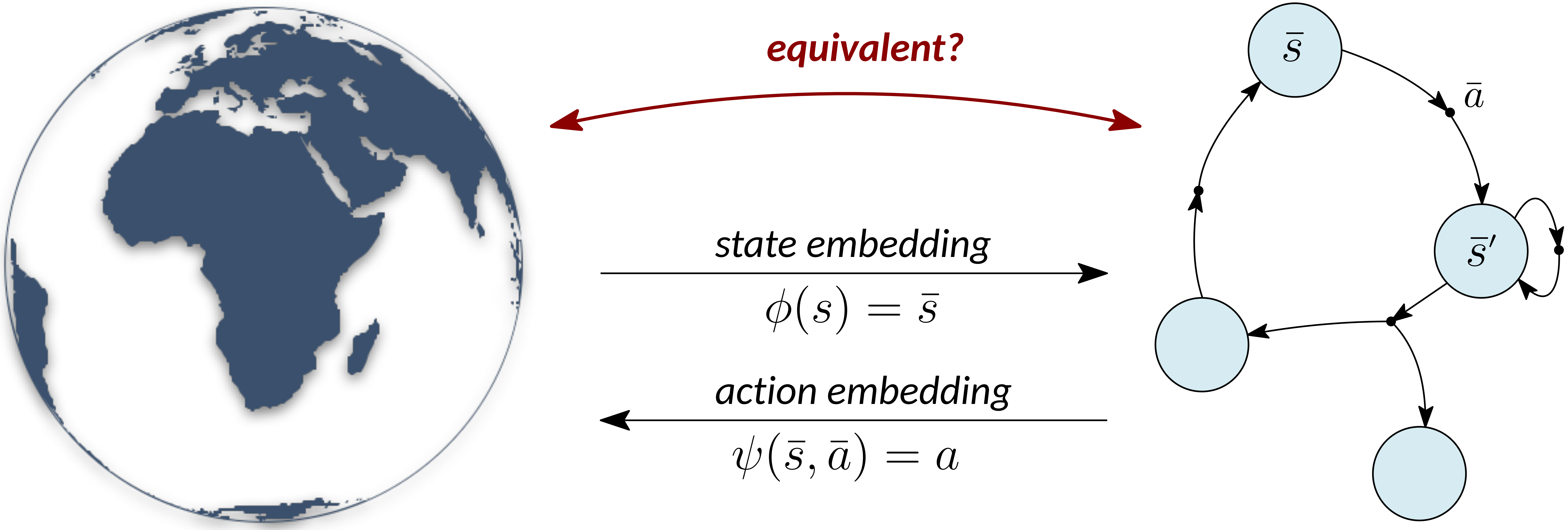
Markov Decision Processes



- State space $\mathscr{S}$
- Action space $\mathscr{A}$
- Reward function $\mathscr{R}: \mathscr{S} \times \mathscr{A} \to \mathbb{R}$
- Probability transition function $\mathbf{P}(s' \mid s, a)$
- Atomic propositions $\mathbf{AP}$ and labelling function $\ell: \mathscr{S} \to 2^{\mathbf{AP}}$

- **Policies** prescribe which action to choose at each step: $\pi: \mathscr{S} \to \Delta(\mathscr{A}), a_t \sim \pi(\cdot \mid s_t)$
- Value functions:
    1. **Discounted return**: $V_\pi(s) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t \cdot \mathscr{R}(s_t, a_t) \mid s_0 = s\right]$
    2. **Properties** φ: $\lim_{\gamma \to \infty} V_\pi(s, \varphi) = \mathbb{P}_\pi(s \models \varphi)$; e.g., $\mathbb{P}_\pi(s \models$ the agent reaches the goal$)$

Bisimulation



Continuous-spaces MDP

Discrete latent MDP

$\mathscr{M} = \langle \mathcal{S}, \mathscr{A}, \mathscr{R}, \mathbf{P}, \ell \rangle$

$\overline{\mathscr{M}} = \langle \overline{\mathcal{S}}, \overline{\mathscr{A}}, \overline{\mathscr{R}}, \overline{\mathbf{P}}, \ell \rangle$

Bisimulation

$B \in \mathcal{S}^2$ is a ***stochastic bisimulation*** iff for all $s_1, s_2 \in \mathcal{S}, a \in \mathcal{A}, T \in \mathcal{S}/B$

$$\ell(s_1) = \ell(s_2) \qquad \mathcal{R}(s_1, a) = \mathcal{R}(s_2, a) \qquad \text{and} \qquad \mathbf{P}(T \mid s_1, a) = \mathbf{P}(T \mid s_2, a)$$

Largest: $\sim$

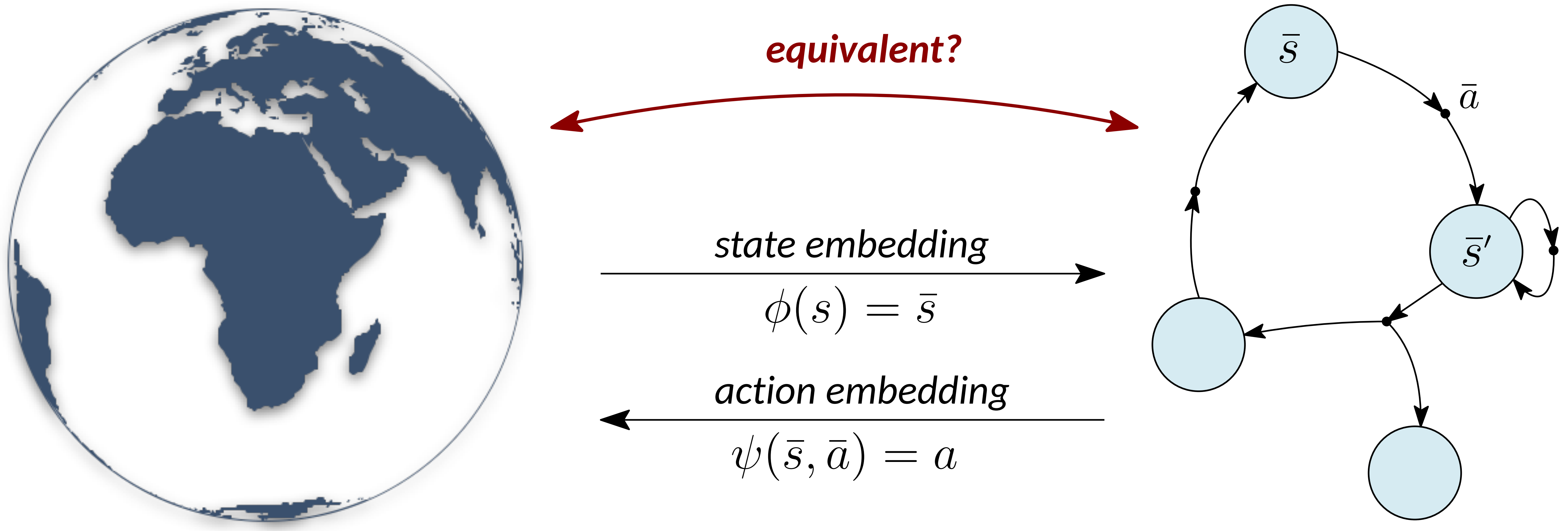
(Larsen and Skou 1989; Givan, Dean, and Greig 2003)

- Behavioral equivalence between states
- Compare two MDPs: take the disjoint union of their state space: $\mathcal{S} \uplus \overline{\mathcal{S}}$

➡ Trajectory, values, and optimal policy equivalence

➡ For a given formal logic $\mathcal{L}$, two bisimilar models satisfy the same set of properties, i.e.,

➡ They *behave the same*



Continuous-spaces MDP

Discrete latent MDP

$\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathbf{P}, \ell \rangle$

$\overline{\mathcal{M}} = \langle \overline{\mathcal{S}}, \overline{\mathcal{A}}, \overline{\mathcal{R}}, \overline{\mathbf{P}}, \ell \rangle$

Bisimulation

$B \in \mathcal{S}^2$ is a ~~stochastic bisimulation iff~~ for all $s_1, s_2 \in \mathcal{S}, a \in \mathcal{A}, T \in \mathcal{S}/B$

$$\ell(s_1) = \ell(s_2) \qquad \mathcal{R}(s_1, a) \neq \mathcal{R}(s_2, a) + \epsilon \quad \text{and} \quad \mathbf{P}(T \mid s_1, a) \neq \mathbf{P}(T \mid s_2, a) + \epsilon$$

Largest: $\sim$

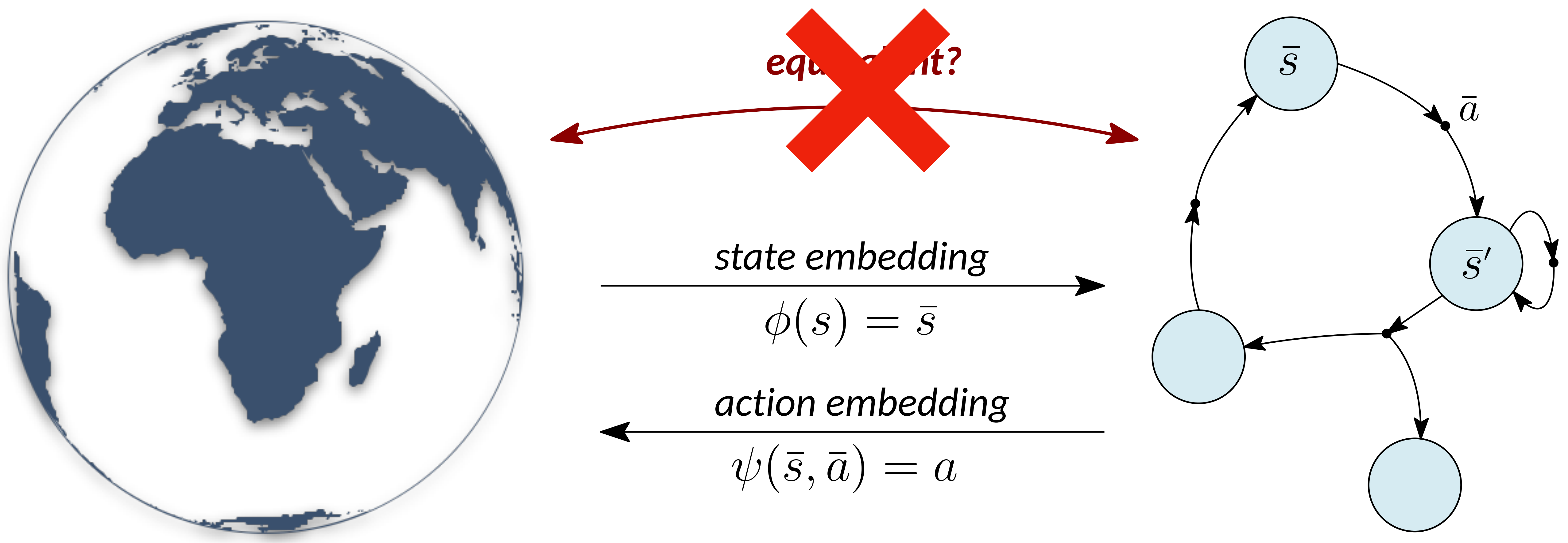
(Larsen and Skou 1989; Givan, Dean, and Greig 2003)

- Behavioral equivalence between states
- Compare two MDPs: take the disjoint union of their state space: $\mathcal{S} \uplus \overline{\mathcal{S}}$

➡ Trajectory, values, and optimal policy equivalence

➡ For a given formal logic $\mathcal{L}$, two bisimilar models satisfy the same set of properties, i.e.,

➡ **They *behave the same***



Continuous-spaces MDP

Discrete latent MDP

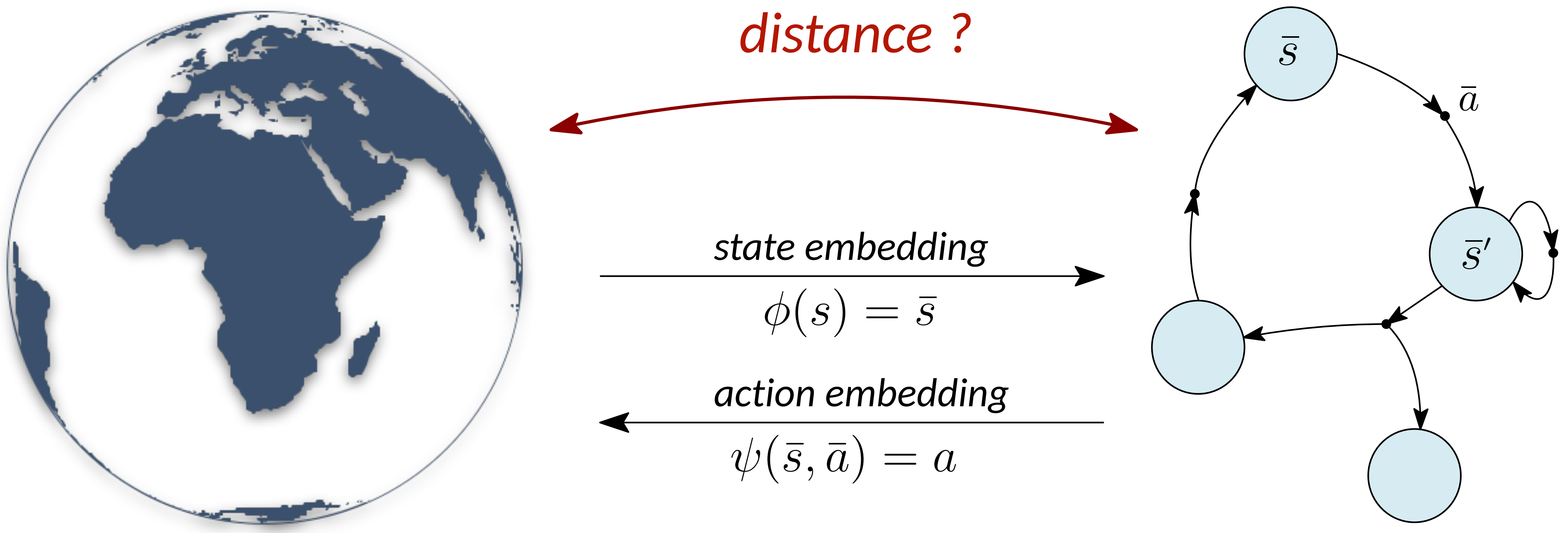
$\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathbf{P}, \ell \rangle$

$\overline{\mathcal{M}} = \langle \overline{\mathcal{S}}, \overline{\mathcal{A}}, \overline{\mathcal{R}}, \overline{\mathbf{P}}, \ell \rangle$

◉ *All or nothing:* two states *nearly identical* with slight numerical difference ϵ are $\neq$

Bisimulation distance

Continuous-spaces MDP — *Discrete latent* MDP



$\mathscr{M} = \langle \mathscr{S}, \mathscr{A}, \mathscr{R}, \mathbf{P}, \ell \rangle$ $\qquad$ $\overline{\mathscr{M}} = \langle \overline{\mathscr{S}}, \overline{\mathscr{A}}, \overline{\mathscr{R}}, \overline{\mathbf{P}}, \ell \rangle$

- For policy π, $\gamma \in [0,1[$, and formal logic $\mathscr{L}$:
  - ➡ ***Bisimulation distance:*** *largest behavioral difference (de Alfaro et. al, 2003; Desharnais et. al, 2004)*

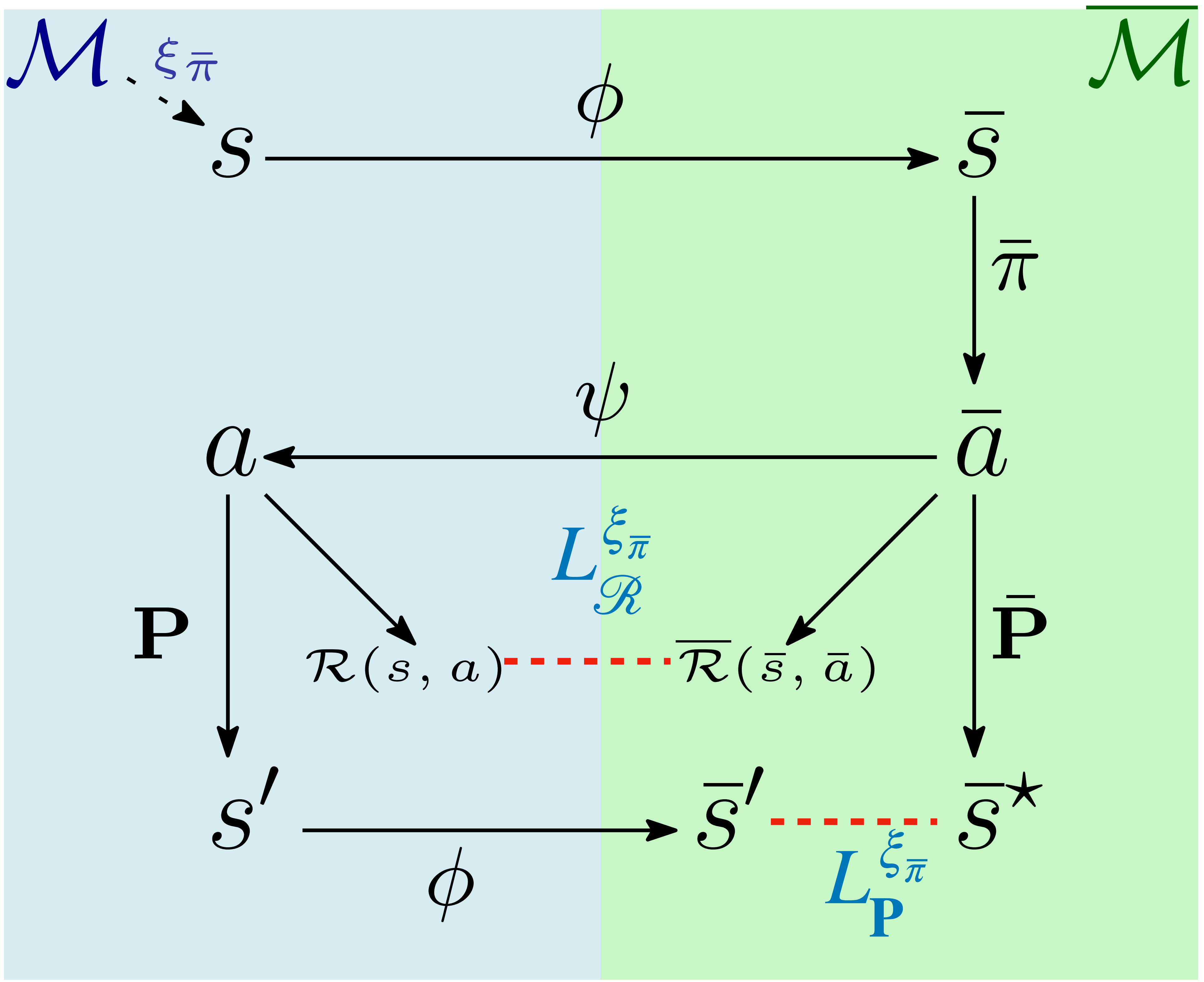
$$\tilde{d}_\pi(s_1, s_2) = \sup_{\varphi \in \mathscr{L}_\gamma} \left| V_\pi(s_1, \varphi) - V_\pi(s_2, \varphi) \right| \quad \forall s_1, s_2 \in \mathscr{S}$$

Take the values of the {event / specification / property}
leading to the largest difference

➡ ***Kernel is bisimilarity:*** $\tilde{d}_\pi(s_1, s_2) = 0 \iff s_1 \sim s_2$

Latent Flow

Execution of a latent policy $\bar{\pi}$ in the original model: **Local Losses**



- Latent policy $\bar{\pi}$, stationary distribution $\xi_{\bar{\pi}}$

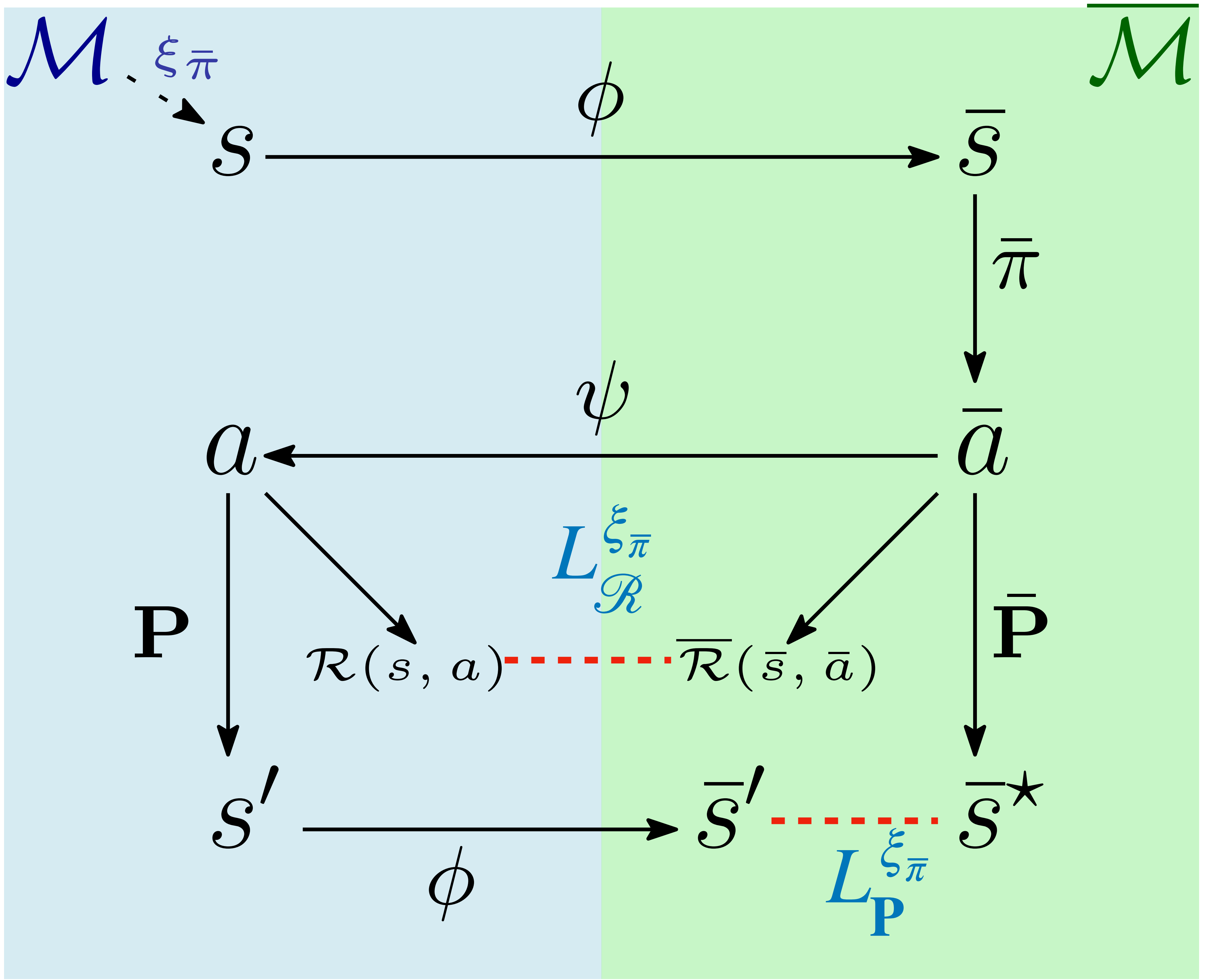
$$L^{\xi_{\bar{\pi}}}_{\mathbf{P}} = \mathbb{E}_{s,\bar{a}\sim\xi_{\bar{\pi}}} W_{d_{\bar{\mathcal{S}}}}\left(\phi\mathbf{P}\left(\cdot \mid s, \bar{a}\right), \overline{\mathbf{P}}\left(\cdot \mid \phi(s), \bar{a}\right)\right)$$

$$L^{\xi_{\bar{\pi}}}_{\mathcal{R}} = \mathbb{E}_{s,\bar{a}\sim\xi_{\bar{\pi}}} \left| \mathcal{R}\left(s, \bar{a}\right) - \overline{\mathcal{R}}\left(\phi(s), \bar{a}\right) \right|$$

- ***Abstraction quality:*** $\mathbb{E}_{s\sim\xi_{\bar{\pi}}} \tilde{d}_{\bar{\pi}}\left(s, \phi(s)\right) \leq \dfrac{L^{\xi_{\bar{\pi}}}_{\mathcal{R}} + \gamma L^{\xi_{\bar{\pi}}}_{\mathbf{P}}}{1-\gamma}$
- ***Representation quality:*** for all $s_1, s_2 \in \mathcal{S}$ such that $\phi(s_1) = \phi(s_2)$

$$\tilde{d}_{\bar{\pi}}\left(s_1, s_2\right) \leq \left(\frac{L^{\xi_{\bar{\pi}}}_{\mathcal{R}} + \gamma L^{\xi_{\bar{\pi}}}_{\mathbf{P}}}{1-\gamma}\right) \cdot \left(\xi^{-1}_{\bar{\pi}}\left(s_1\right) + \xi^{-1}_{\bar{\pi}}\left(s_2\right)\right)$$

Execution of a latent policy $\bar{\pi}$ in the original model: **Local Losses**



- Latent policy $\bar{\pi}$, stationary distribution $\xi_{\bar{\pi}}$

$$L_{\mathbf{P}}^{\xi_{\bar{\pi}}} = \mathbb{E}_{s,\bar{a}\sim\xi_{\bar{\pi}}} W_{d_{\bar{\mathcal{S}}}}\left(\phi\mathbf{P}\left(\,\cdot \mid s, \bar{a}\right), \bar{\mathbf{P}}\left(\,\cdot \mid \phi(s), \bar{a}\right)\right)$$

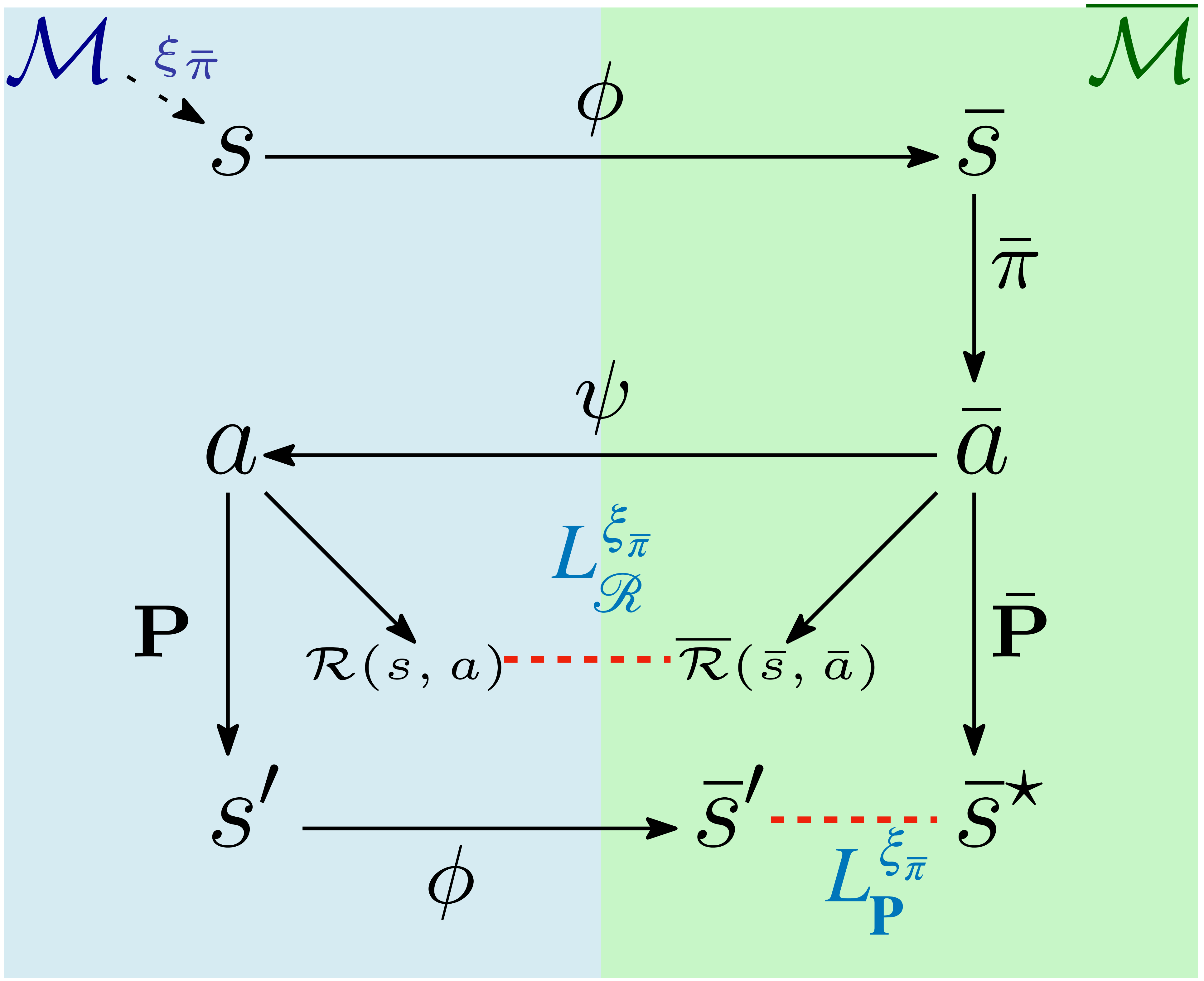
$$L_{\mathscr{R}}^{\xi_{\bar{\pi}}} = \mathbb{E}_{s,\bar{a}\sim\xi_{\bar{\pi}}} \left| \mathscr{R}\left(s, \bar{a}\right) - \overline{\mathscr{R}}\left(\phi(s), \bar{a}\right) \right|$$

- ***Abstraction quality:*** $\mathbb{E}_{s\sim\xi_{\bar{\pi}}} \left| V_{\bar{\pi}}(s) - \bar{V}_{\bar{\pi}}(s) \right| \leq \dfrac{L_{\mathscr{R}}^{\xi_{\bar{\pi}}} + \gamma L_{\mathbf{P}}^{\xi_{\bar{\pi}}}}{1-\gamma}$
- ***Representation quality:*** for all $s_1, s_2 \in \mathcal{S}$ such that $\phi(s_1) = \phi(s_2)$

$$\left| V_{\bar{\pi}}(s_1) - V_{\bar{\pi}}(s_2) \right| \leq \left(\frac{L_{\mathscr{R}}^{\xi_{\bar{\pi}}} + \gamma L_{\mathbf{P}}^{\xi_{\bar{\pi}}}}{1-\gamma} \right) \cdot \left(\xi_{\bar{\pi}}^{-1}\left(s_1\right) + \xi_{\bar{\pi}}^{-1}\left(s_2\right) \right)$$

Latent Flow

Execution of a latent policy $\bar\pi$ in the original model: **Local Losses**



- Latent policy $\bar\pi$, stationary distribution $\xi_{\bar\pi}$

$$L_{\mathbf{P}}^{\xi_{\bar\pi}} = \mathbb{E}_{s,\bar a\sim\xi_{\bar\pi}} W_{d_{\bar{\mathcal{S}}}}\left(\phi\mathbf{P}\left(\cdot \mid s, \bar a\right), \overline{\mathbf{P}}\left(\cdot \mid \phi(s), \bar a\right)\right)$$

$$L_{\mathcal{R}}^{\xi_{\bar\pi}} = \mathbb{E}_{s,\bar a\sim\xi_{\bar\pi}} \left|\mathcal{R}\left(s, \bar a\right) - \overline{\mathcal{R}}\left(\phi(s), \bar a\right)\right|$$

- ***Abstraction quality**:* $\mathbb{E}_{s\sim\xi_{\bar\pi}}\left|V_{\bar\pi}(s) - \bar V_{\bar\pi}(s)\right| \leq \dfrac{L_{\mathcal{R}}^{\xi_{\bar\pi}} + \gamma L_{\mathbf{P}}^{\xi_{\bar\pi}}}{1-\gamma}$
- ***Representation quality**:* for all $s_1, s_2 \in \mathcal{S}$ such that $\phi(s_1) = \phi(s_2)$

$$\left|V_{\bar\pi}(s_1) - V_{\bar\pi}(s_2)\right| \leq \left(\frac{L_{\mathcal{R}}^{\xi_{\bar\pi}} + \gamma L_{\mathbf{P}}^{\xi_{\bar\pi}}}{1-\gamma}\right) \cdot \left(\xi_{\bar\pi}^{-1}\left(s_1\right) + \xi_{\bar\pi}^{-1}\left(s_2\right)\right)$$

- **PAC scheme from samples**: let *trace* $\langle s_{0:T}, \bar a_{0:T-1}, r_{0:T-1}\rangle \sim \xi_{\bar\pi}$, $\epsilon, \delta \in]0,1[$ and

$$T \geq \left\lceil \frac{-\log\left(\delta/4\right)}{2\epsilon^2} \right\rceil:$$

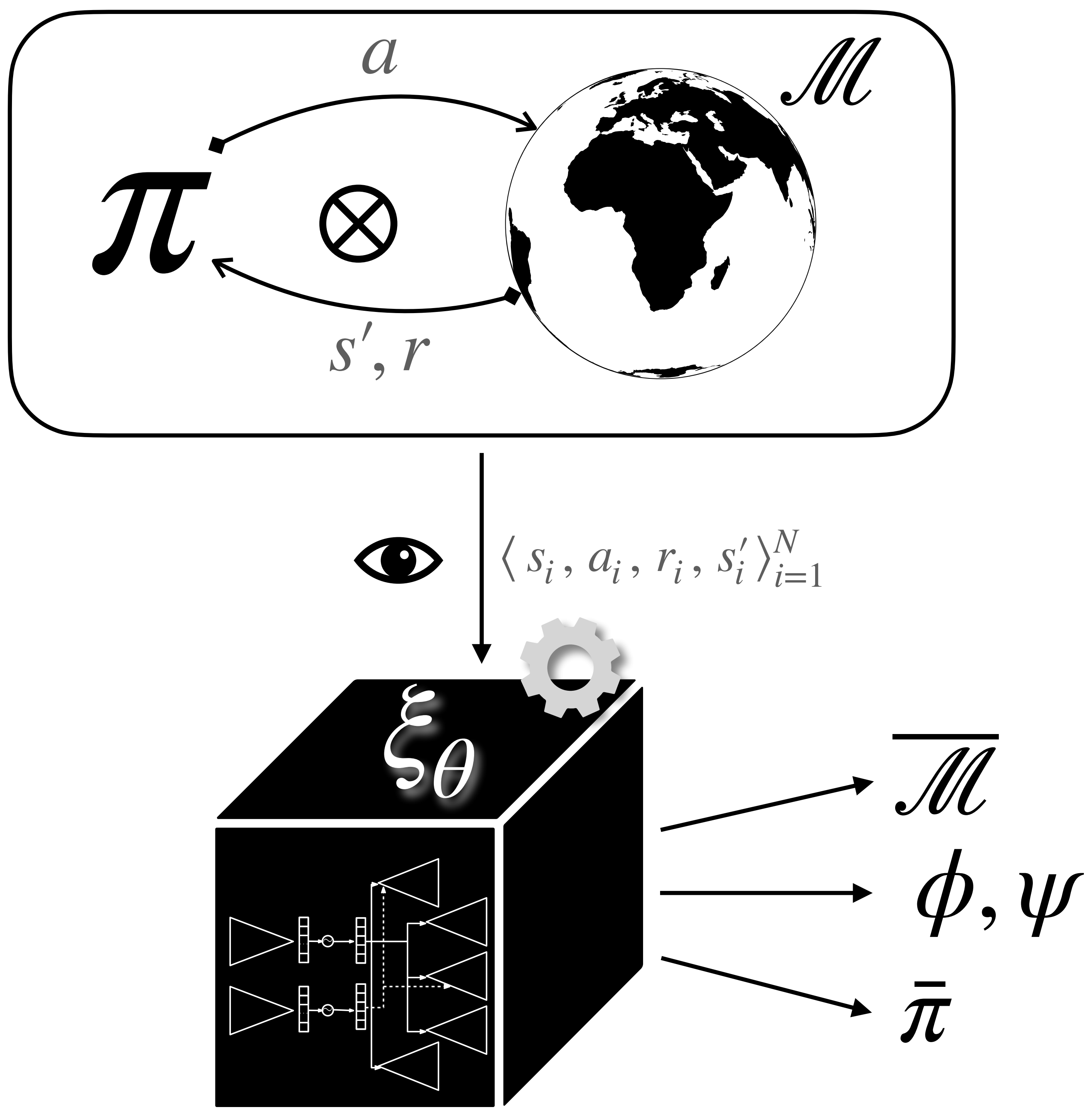
$$\hat L_{\mathcal{R}}^{\xi_{\bar\pi}} = \frac{1}{T}\sum_{t=0}^{T-1}\left|r_t - \overline{\mathcal{R}}\left(\phi(s_t), \bar a_t\right)\right| \quad \text{and} \quad \hat L_{\mathbf{P}}^{\xi_{\bar\pi}} = \frac{1}{T}\sum_{t=0}^{T-1}\left[1 - \overline{\mathbf{P}}\left(\phi(s_{t+1}) \mid \phi(s_t), \bar a_t\right)\right]$$

Then, $\left|L_{\mathcal{R}}^{\xi_{\bar\pi}} - \hat L_{\mathcal{R}}^{\xi_{\bar\pi}}\right| \leq \epsilon$ and $\left|L_{\mathbf{P}}^{\xi_{\bar\pi}} - \hat L_{\mathbf{P}}^{\xi_{\bar\pi}}\right| \leq \epsilon$ **with probability $1-\delta$**

Learning the Latent Space Model

- Train a *behavioral model* ξ_θ by learning from traces produced by executing the RL policy π in the original model $\mathcal{M}$
- Goal: learn ξ_θ so that we can retrieve:
  - The latent MDP $\overline{\mathcal{M}} = \langle \overline{\mathcal{S}}, \overline{\mathcal{A}}, \overline{\mathcal{R}}, \overline{\mathbf{P}}, \ell \rangle$
  - The embedding functions ϕ, ψ
  - A latent policy $\bar{\pi}$ distilled from π
- Minimize a *discrepancy* D between $\mathcal{M} \otimes \pi$ and ξ_θ

$$\min_{\theta} D_{KL}\left(\mathcal{M} \otimes \pi, \xi_\theta\right)$$

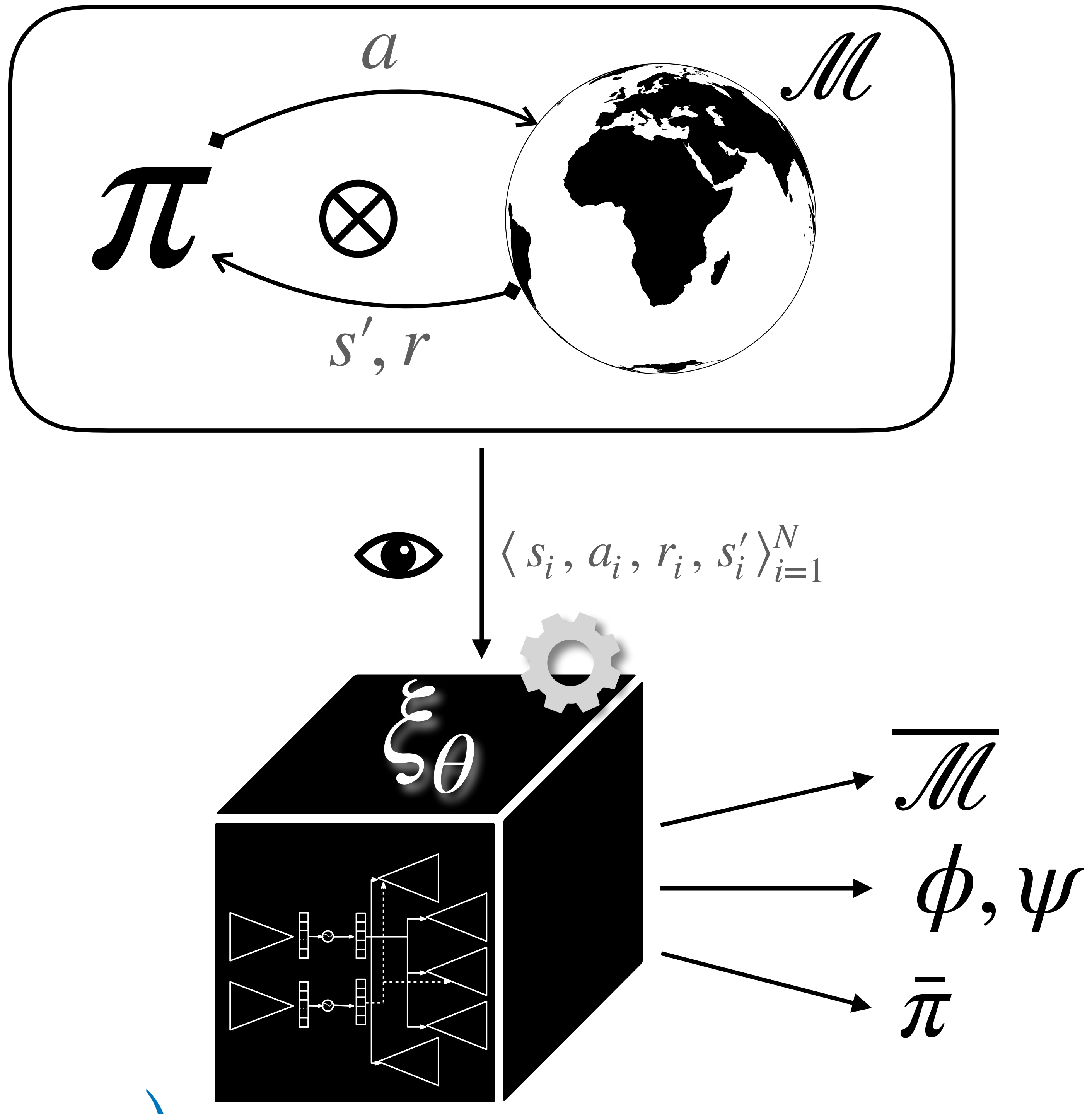


- Choose the *Kullback-Leibler divergence*

$$D_{KL}\left(P, Q\right) = \mathbb{E}_{x \sim P}\left[\log\left(\frac{P(x)}{Q(x)}\right)\right]$$

Learning the Latent Space Model

- Train a *behavioral model* ξ_θ by learning from traces produced by executing the RL policy π in the original model $\mathcal{M}$
- Goal: learn ξ_θ so that we can retrieve:
  - The latent MDP $\overline{\mathcal{M}} = \langle \overline{\mathcal{S}}, \overline{\mathcal{A}}, \overline{\mathcal{R}}, \overline{\mathbf{P}}, \ell \rangle$
  - The embedding functions ϕ, ψ
  - A latent policy $\bar{\pi}$ distilled from π
- Minimize a *discrepancy* D between $\mathcal{M} \otimes \pi$ and ξ_θ



$$\min_\theta D_{KL}\left(\mathcal{M} \otimes \pi, \xi_\theta\right)$$

$$\equiv \max_\theta \mathbb{E}_{\tau \sim \mathcal{M} \otimes \pi}\left[\log \xi_\theta(\tau)\right] \geq \max_{\iota, \theta} ELBO\left(\overline{\mathcal{M}}_\theta, \phi_\iota, \psi_\theta\right)$$

(Kingma & Welling, 2014; Hoffman et al., 2013)

- Choose the *Kullback-Leibler divergence*

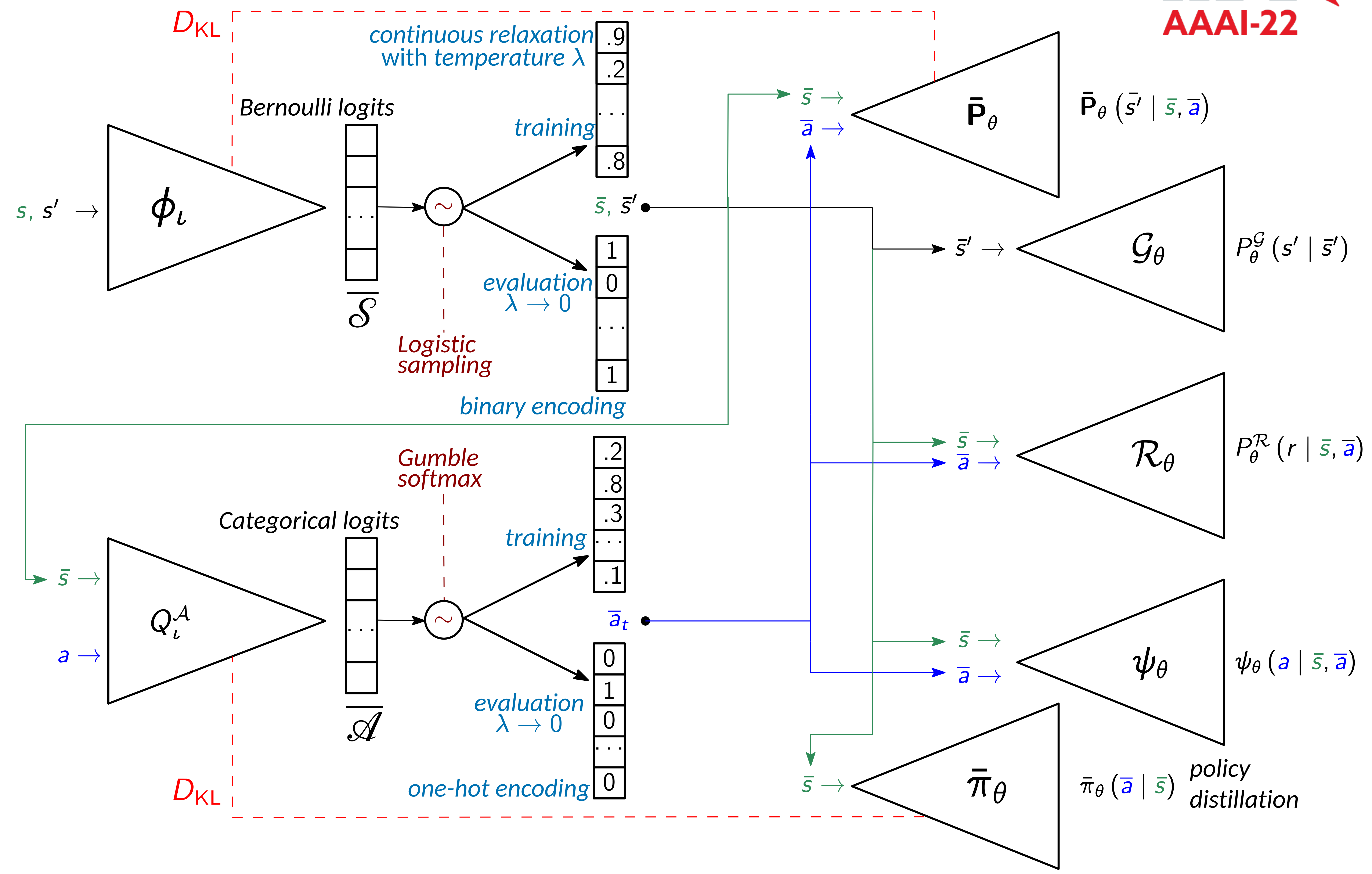
$$D_{KL}\left(P, Q\right) = \mathbb{E}_{x \sim P}\left[\log\left(\frac{P(x)}{Q(x)}\right)\right]$$

Variational Markov Decision Process

$$\max_{\iota,\theta} \; ELBO\left(\overline{\mathcal{M}}_{\theta}, \phi_{\iota}, \psi_{\theta}\right) = -\min_{\iota,\theta} \left\{\mathbf{D}_{\iota,\theta} + \mathbf{R}_{\iota,\theta}\right\}$$

$$\mathbf{D}_{\iota,\theta} = - \mathop{\mathbb{E}}_{\substack{s,a,r,s' \sim \xi_{\pi} \\ \bar{s},\bar{s}' \sim \phi_{\iota}(\cdot \mid s,s') \\ \bar{a} \sim Q_{\iota}^{\mathcal{A}}(\cdot \mid \bar{s},a)}} \left[\log P_{\theta}^{\mathcal{G}}(s' \mid \bar{s}') + \log \psi_{\theta}(a \mid \bar{s}, \bar{a}) + \log P_{\theta}^{\mathcal{R}}(r \mid \bar{s}, \bar{a}) \right]$$

$$\mathbf{R}_{\iota,\theta} = \mathop{\mathbb{E}}_{\substack{s,a,s' \sim \xi_{\pi} \\ \bar{s} \sim \phi_{\iota}(\cdot \mid s) \\ \bar{a} \sim Q_{\iota}^{\mathcal{A}}(\cdot \mid \bar{s},a)}} \left[D_{\mathrm{KL}}\left(\phi_{\iota}(\cdot \mid s') \,\|\, \bar{\mathbf{P}}_{\theta}(\cdot \mid \bar{s}, \bar{a})\right) + D_{\mathrm{KL}}\left(Q_{\iota}^{\mathcal{A}}(\cdot \mid \bar{s}, a) \,\|\, \bar{\pi}_{\theta}(\cdot \mid \bar{s})\right) \right]$$



Variational Markov Decision Process

$$\max_{\iota,\theta} \mathit{ELBO}\left(\overline{\mathcal{M}}_\theta, \phi_\iota, \psi_\theta\right) = -\min_{\iota,\theta}\left\{\mathbf{D}_{\iota,\theta} + \mathbf{R}_{\iota,\theta}\right\}$$

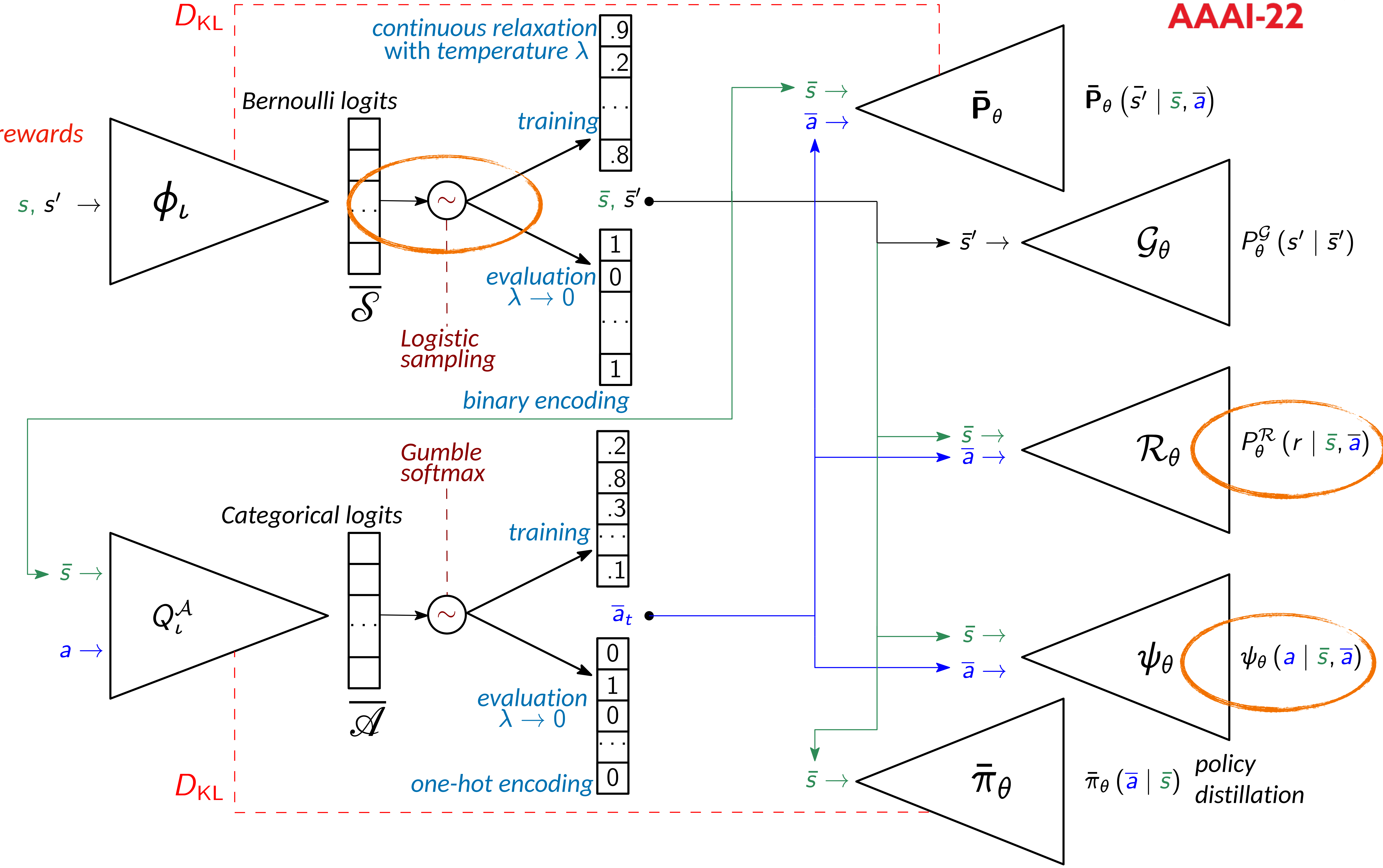
$$\mathbf{D}_{\iota,\theta} = -\mathop{\mathbb{E}}_{\substack{s,a,r,s' \sim \xi_\pi \\ \bar{s},\bar{s}' \sim \phi_\iota(\cdot \mid s,s') \\ \bar{a} \sim Q^{\mathcal{A}}_\iota(\cdot \mid \bar{s},a)}} \left[\log P^{\mathcal{G}}_\theta(s' \mid \bar{s}') + \log \psi_\theta(a \mid \bar{s}, \bar{a}) + \log P^{\mathcal{R}}_\theta(r \mid \bar{s}, \bar{a})\right]$$

Log-likelihood of rewards

$$\mathbf{R}_{\iota,\theta} = \mathop{\mathbb{E}}_{\substack{s,a,s' \sim \xi_\pi \\ \bar{s} \sim \phi_\iota(\cdot \mid s) \\ \bar{a} \sim Q^{\mathcal{A}}_\iota(\cdot \mid \bar{s},a)}} \left[D_{KL}\left(\phi_\iota(\cdot \mid s') \parallel \bar{\mathbf{P}}_\theta(\cdot \mid \bar{s}, \bar{a})\right) + D_{\mathrm{KL}}\left(Q^{\mathcal{A}}_\iota(\cdot \mid \bar{s}, a) \parallel \bar{\pi}_\theta(\cdot \mid \bar{s})\right)\right]$$

$$L^{\xi_\pi}_{\mathbf{P}} = \mathbb{E}_{s,\bar{a} \sim \xi_\pi} W_{d_{\bar{\mathcal{S}}}}\left(\phi\mathbf{P}\left(\cdot \mid s, \bar{a}\right), \bar{\mathbf{P}}\left(\cdot \mid \phi(s), \bar{a}\right)\right)$$
$$\leq \mathbb{E}_{s,\bar{a},s' \sim \xi_\pi} W_{d_{\bar{\mathcal{S}}}}\left(\phi\left(\cdot \mid s'\right), \bar{\mathbf{P}}\left(\cdot \mid \phi(s), \bar{a}\right)\right)$$

- Stochastic embedding and reward functions
  ➡ Determinized after the learning process
- Variational proxies to local losses



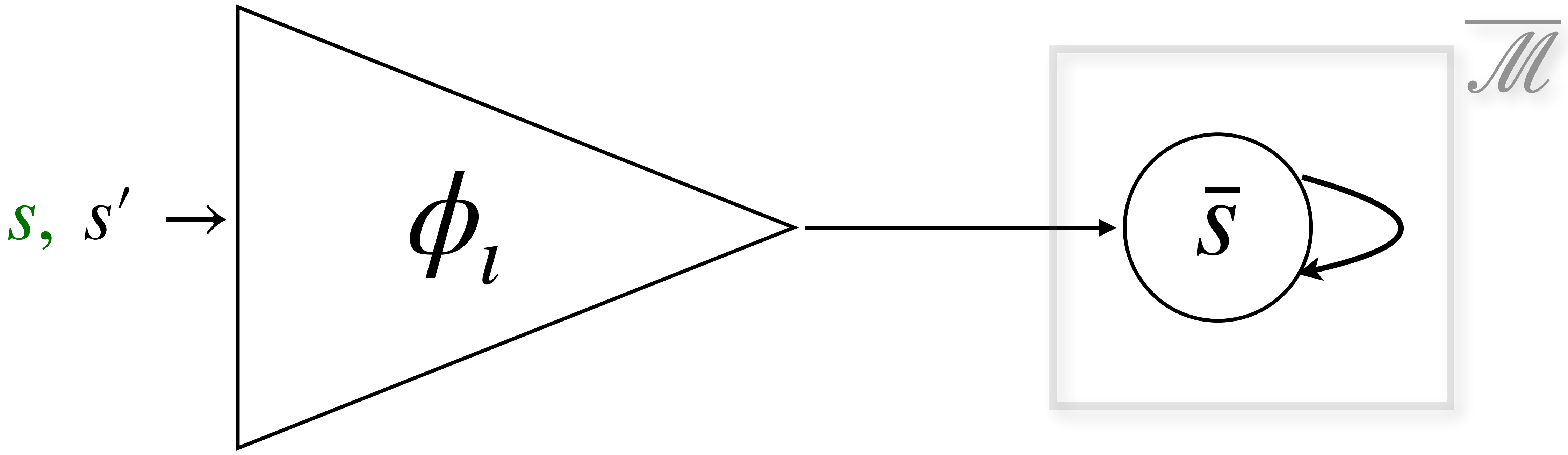
Variational Markov Decision Process

$$\max_{\iota,\theta} ELBO\left(\overline{\mathcal{M}}_\theta, \phi_\iota, \psi_\theta\right) = -\min_{\iota,\theta}\left\{\mathbf{D}_{\iota,\theta} + \mathbf{R}_{\iota,\theta}\right\}$$

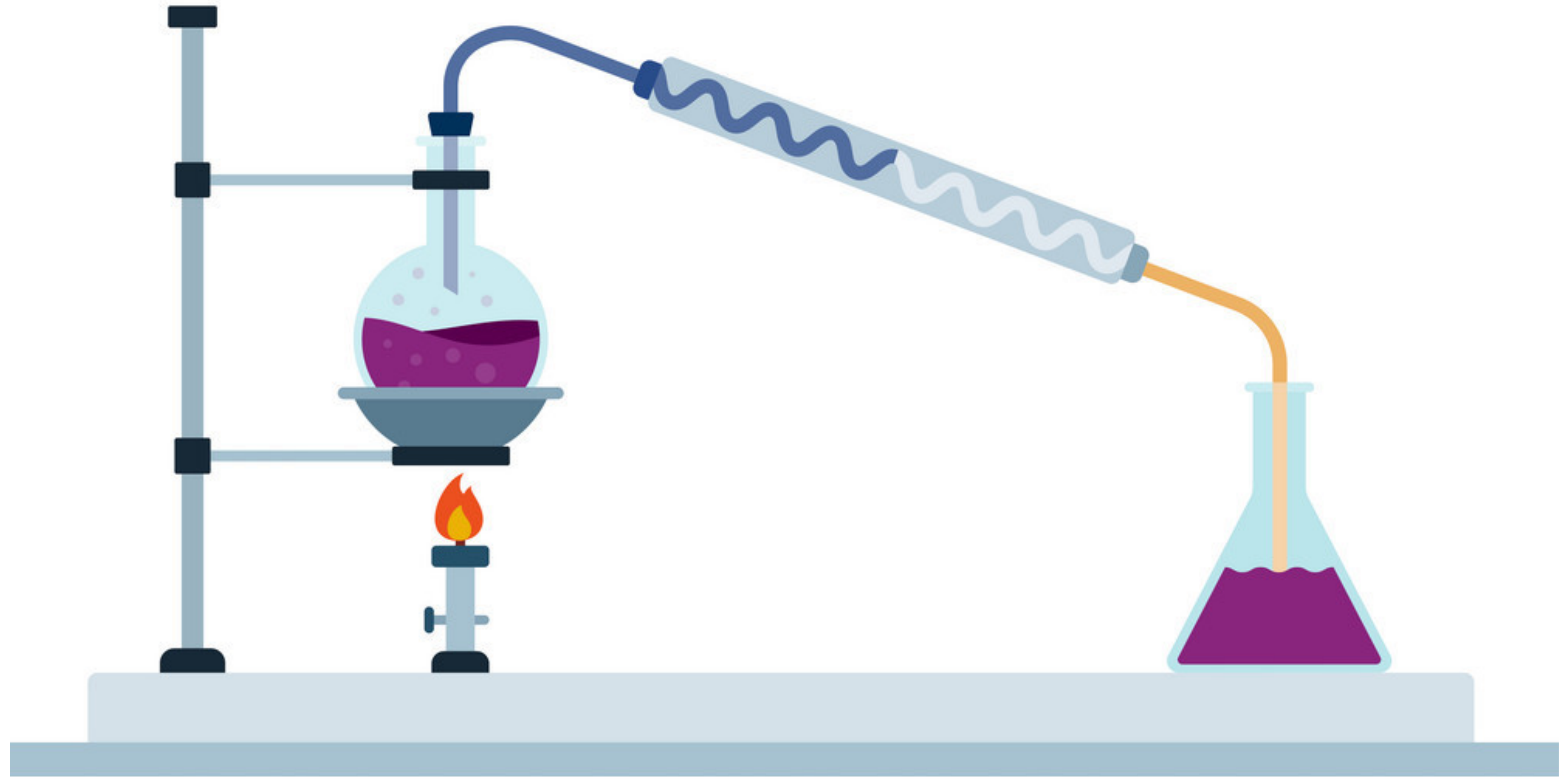
$$\mathbf{D}_{\iota,\theta} = -\mathop{\mathbb{E}}_{\substack{s,a,r,s'\sim\xi_\pi \\ \bar{s},\bar{s}'\sim\phi_\iota(\cdot\mid s,s') \\ \bar{a}\sim Q^{\mathcal{A}}_\iota(\cdot\mid\bar{s},a)}}\left[\log P^{\mathcal{G}}_\theta(s'\mid\bar{s}') + \log\psi_\theta(a\mid\bar{s},\bar{a}) + \log P^{\mathcal{R}}_\theta(r\mid\bar{s},\bar{a})\right]$$

$$\mathbf{R}_{\iota,\theta} = \mathop{\mathbb{E}}_{\substack{s,a,s'\sim\xi_\pi \\ \bar{s}\sim\phi_\iota(\cdot\mid s) \\ \bar{a}\sim Q^{\mathcal{A}}_\iota(\cdot\mid\bar{s},a)}}\left[D_{KL}\left(\phi_\iota(\cdot\mid s')\parallel\bar{\mathbf{P}}_\theta(\cdot\mid\bar{s},\bar{a})\right) + D_{\mathrm{KL}}\left(Q^{\mathcal{A}}_\iota(\cdot\mid\bar{s},a)\parallel\bar{\pi}_\theta(\cdot\mid\bar{s})\right)\right]$$

$$L^{\xi_\pi}_{\mathbf{P}} = \mathbb{E}_{s,\bar{a}\sim\xi_\pi} W_{d_{\bar{S}}}\left(\phi\mathbf{P}(\cdot\mid s,\bar{a}), \bar{\mathbf{P}}(\cdot\mid\phi(s),\bar{a})\right)$$
$$\leq \mathbb{E}_{s,\bar{a},s'\sim\xi_\pi} W_{d_{\bar{S}}}\left(\phi(\cdot\mid s'), \bar{\mathbf{P}}(\cdot\mid\phi(s),\bar{a})\right)$$

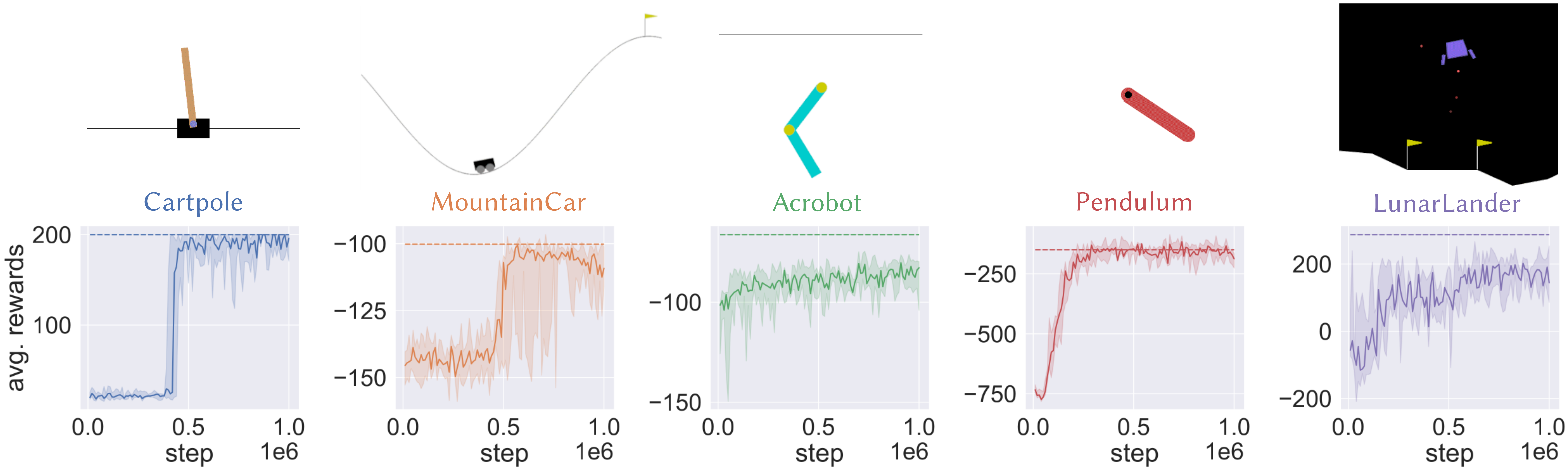


- Stochastic embedding and reward functions
  ➡ Determinized after the learning process
- Variational proxies to local losses
  ➡ **Posterior collapse**
  ➡ **fix: prioritized replay buffers, entropy regularization, annealing scheme**



VAE-MDP: Evaluation

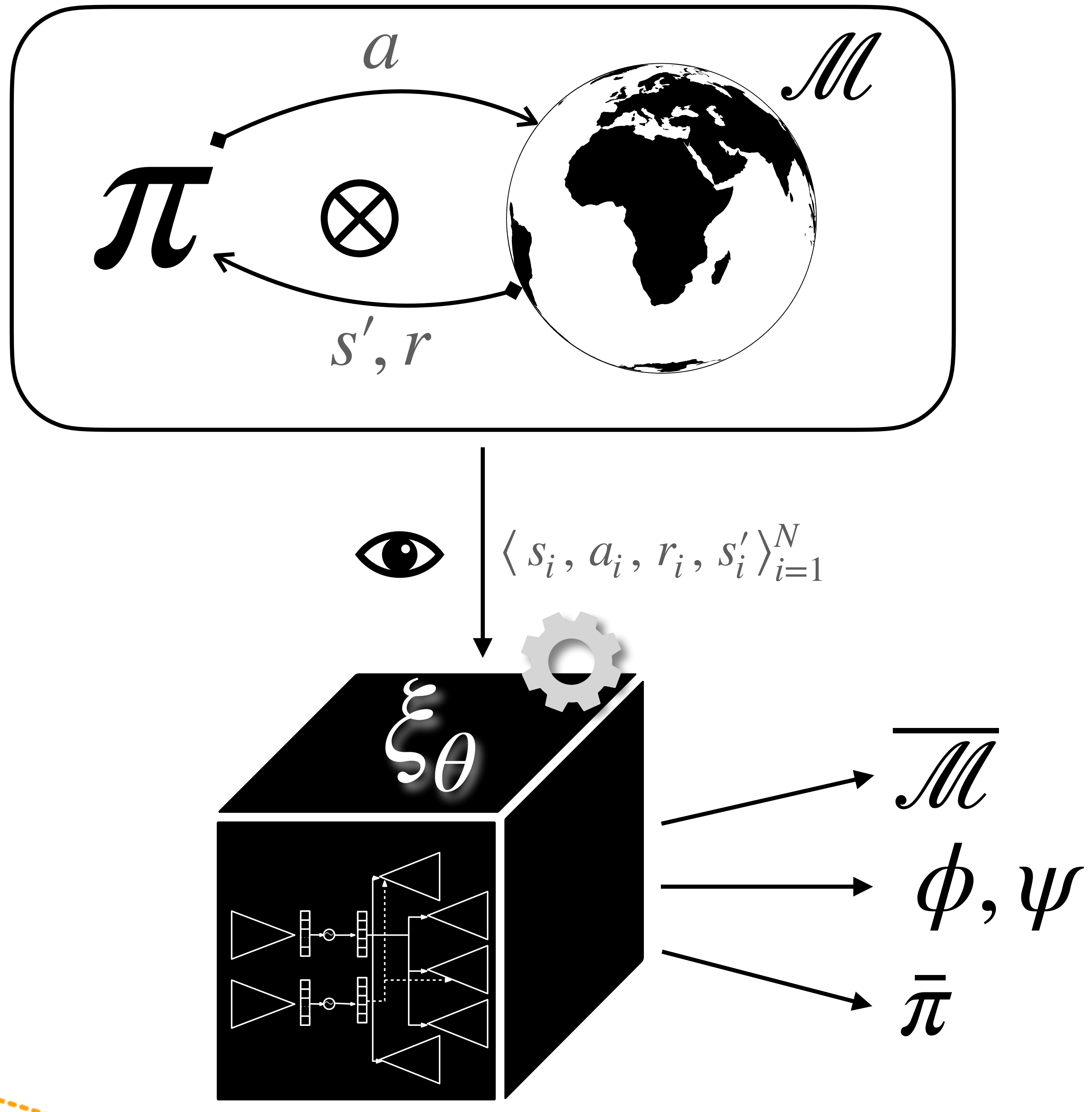
Distillation: performance of $\bar{\pi}$



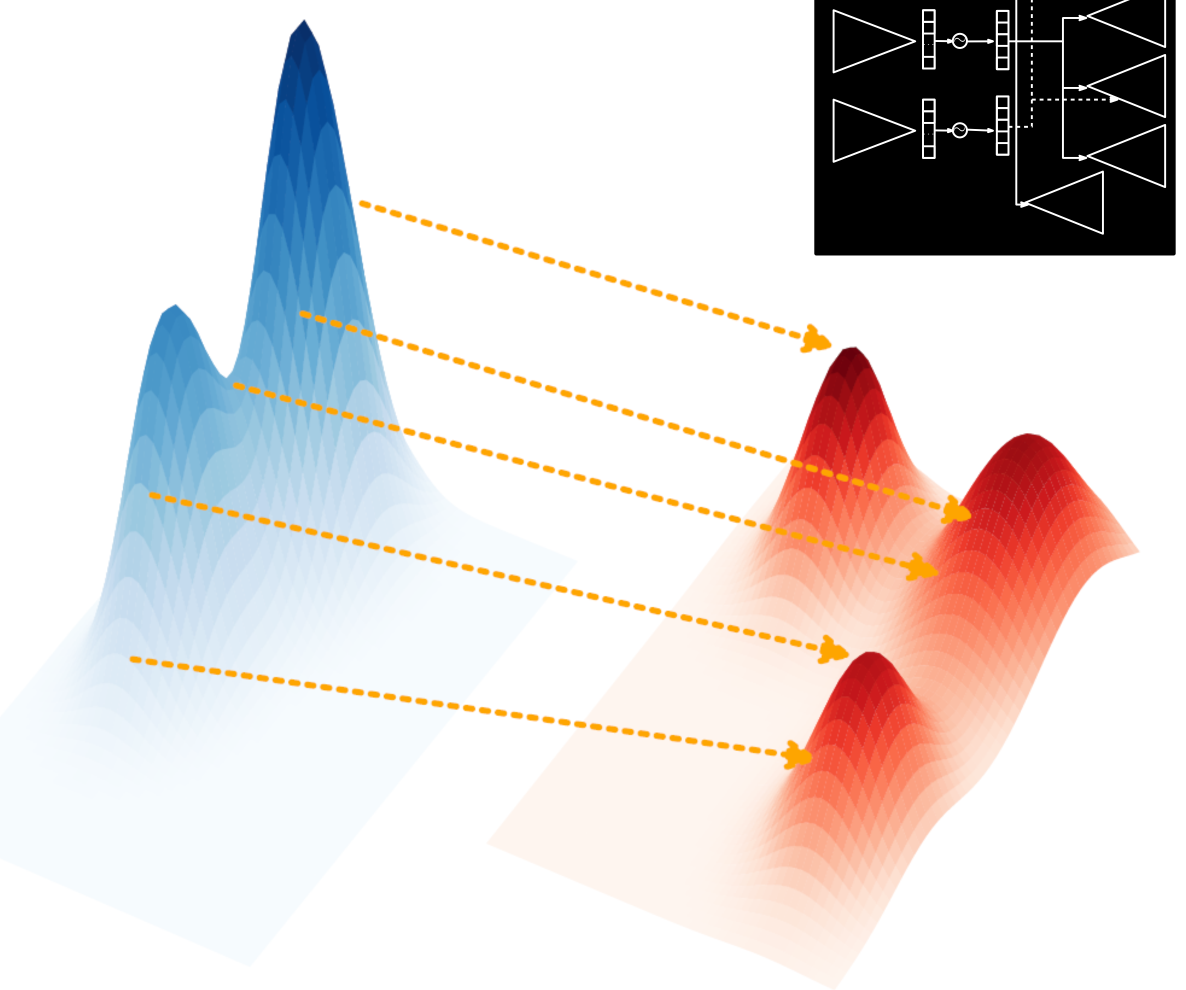
Handling posterior collapse slows down the learning process

Learning the Latent Space Model

- Train a *behavioral model* ξ_θ by learning from traces produced by executing the RL policy π in the original model $\mathcal{M}$
- Goal: learn ξ_θ so that we can retrieve:
  - The latent MDP $\overline{\mathcal{M}} = \langle \overline{\mathcal{S}}, \overline{\mathcal{A}}, \overline{\mathcal{R}}, \overline{\mathbf{P}}, \ell \rangle$
  - The embedding functions ϕ, ψ
  - A latent policy $\bar{\pi}$ distilled from π
- Minimize a *discrepancy* D between $\mathcal{M} \otimes \pi$ and ξ_θ



$$\min_\theta W\left(\mathcal{M} \otimes \pi, \xi_\theta\right)$$

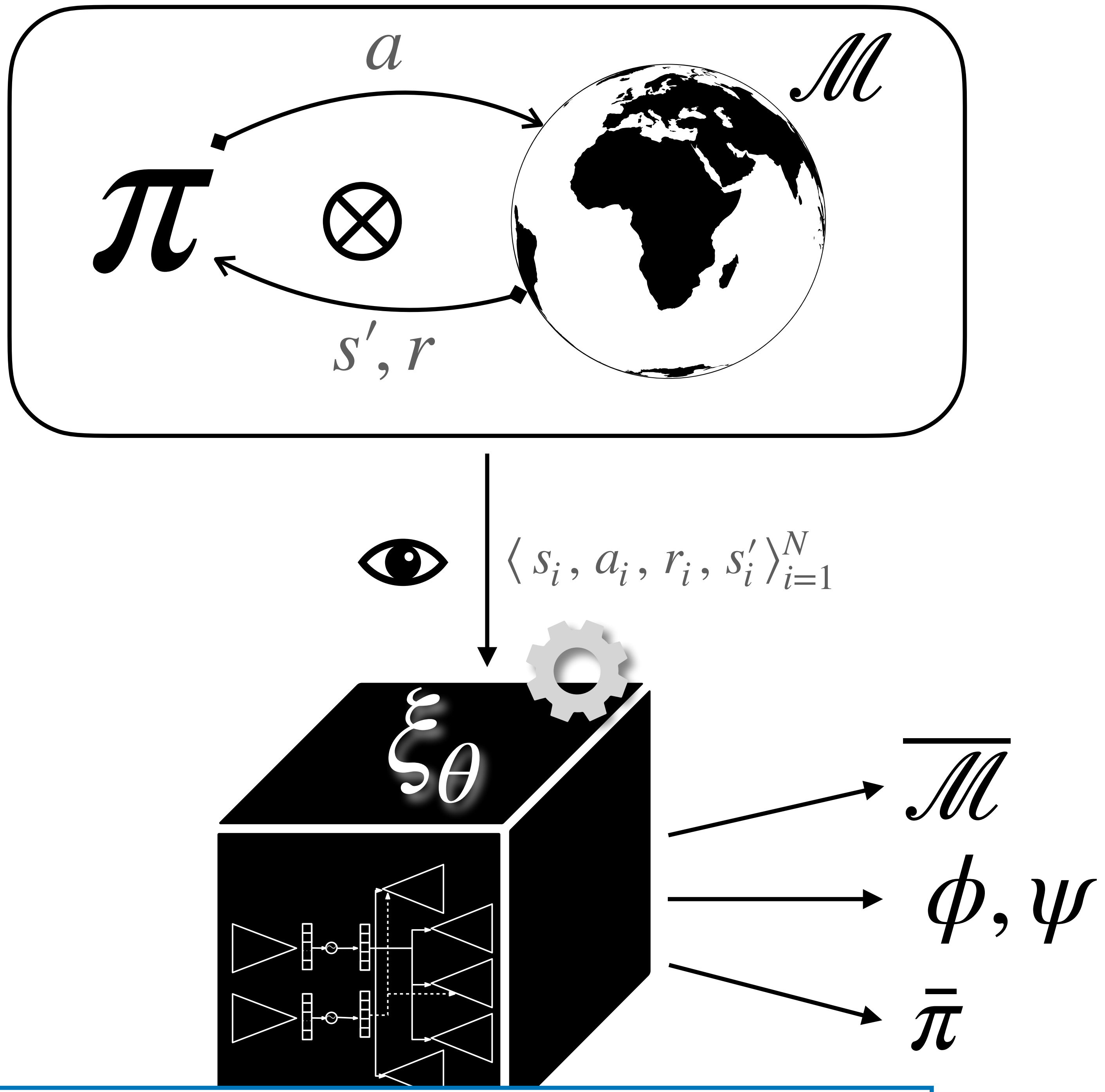


- Choose the *Wasserstein Distance*

$$W\left(P, Q\right) = \inf_{\lambda \in \Lambda(P,Q)} \mathbb{E}_{x,y \sim \lambda}\, d\left(x, y\right) = \sup_{\|f\| \leq 1} \mathbb{E}_{x \sim P} f(x) - \mathbb{E}_{y \sim Q} f\left(y\right)$$

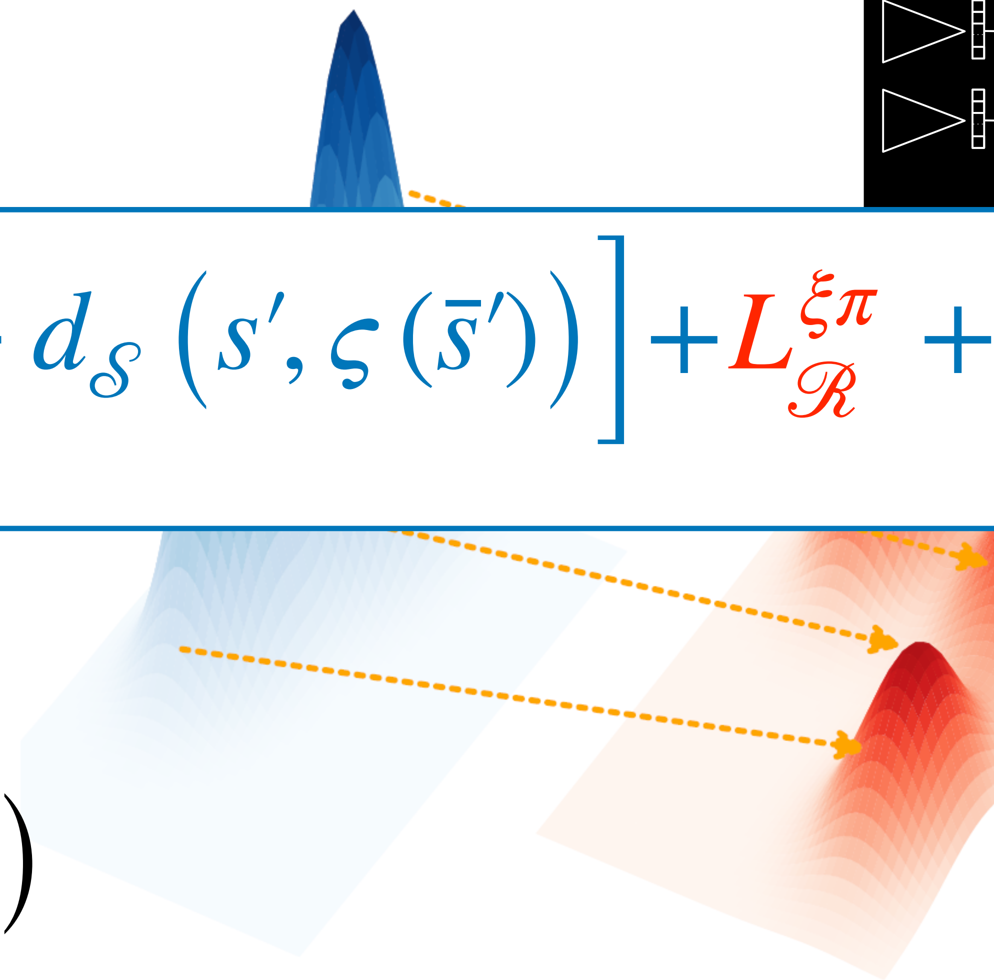
Learning the Latent Space Model

- Train a *behavioral model* ξ_θ by learning from traces produced by executing the RL policy π in the original model $\mathcal{M}$
- Goal: learn ξ_θ so that we can retrieve:
  - The latent MDP $\overline{\mathcal{M}} = \langle \overline{\mathcal{S}}, \overline{\mathcal{A}}, \overline{\mathcal{R}}, \overline{\mathbf{P}}, \ell \rangle$
  - The embedding functions ϕ, ψ
  - A latent policy $\bar{\pi}$ distilled from π
- Minimize a *discrepancy* D between $\mathcal{M} \otimes \pi$ and ξ_θ



$$\min_\theta W\left(\mathcal{M} \otimes \pi, \xi_\theta\right)$$

$$\leq \min \mathbb{E}_{s,a,s' \sim \xi_\pi} \mathbb{E}_{\bar{s},\bar{a},\bar{s}' \sim \phi(\,\cdot\mid s,a,s')} \left[d_{\mathcal{S}}\left(s, \varsigma\left(\bar{s}\right)\right) + d_{\mathcal{A}}\left(a, \psi\left(\bar{s}, \bar{a}\right)\right) + d_{\mathcal{S}}\left(s', \varsigma\left(\bar{s}'\right)\right) \right] + L_{\mathcal{R}}^{\xi\pi} + \beta\left(\mathcal{W}_{\xi_\pi} + L_{\mathbf{P}}^{\xi_\pi}\right)$$

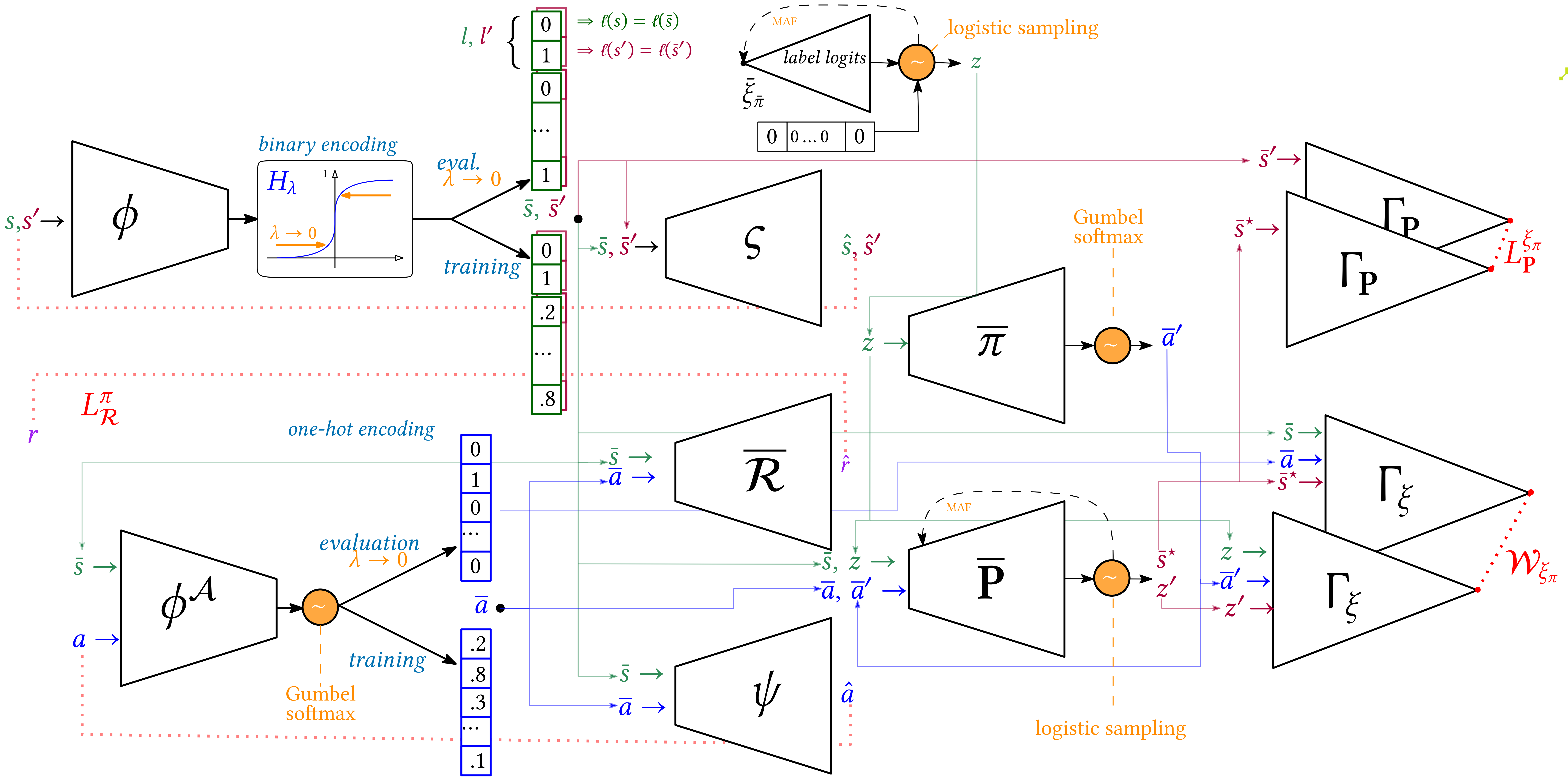


- Choose the *Wasserstein Distance*

$$W\left(P, Q\right) = \inf_{\lambda \in \Lambda(P,Q)} \mathbb{E}_{x,y \sim \lambda}\, d\left(x, y\right) = \sup_{\|f\| \leq 1} \mathbb{E}_{x \sim P}\, f(x) - \mathbb{E}_{y \sim Q}\, f\left(y\right)$$

Wasserstein Auto-encoded Markov Decision Process

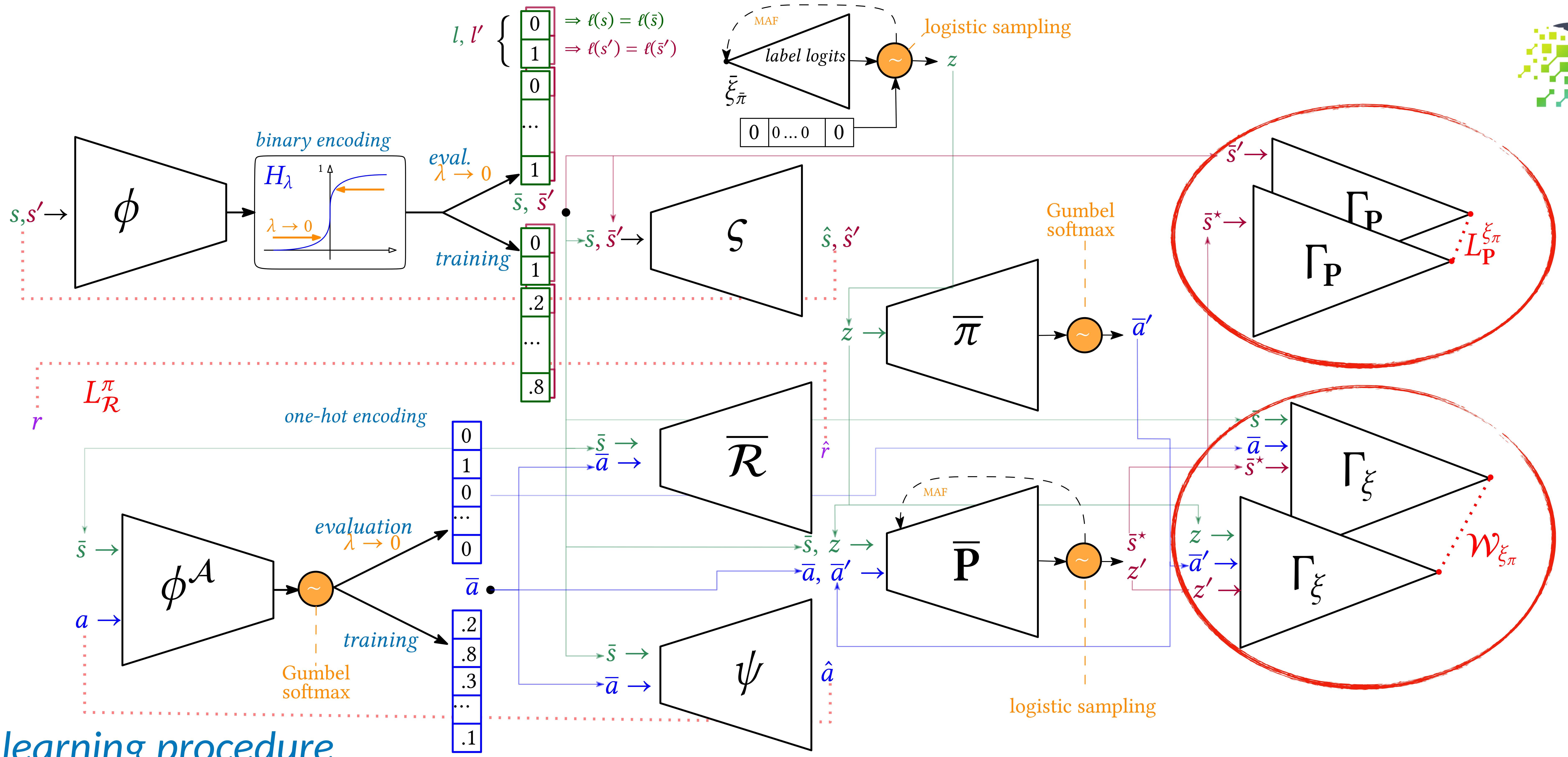
$$\min \mathbb{E}_{s,a,s'\sim\xi_\pi}\mathbb{E}_{\bar{s},\bar{a},\bar{s}'\sim\phi(\,\cdot\mid s,a,s')}\left[d_{\mathcal{S}}\left(s,\varsigma\left(\bar{s}\right)\right)+d_{\mathcal{A}}\left(a,\psi\left(\bar{s},\bar{a}\right)\right)+d_{\mathcal{S}}\left(s',\varsigma\left(\bar{s}'\right)\right)\right]+L_{\mathcal{R}}^{\xi\pi}+\beta\left(\mathcal{W}_{\xi_\pi}+L_{\mathbf{P}}^{\xi_\pi}\right)$$



- $\mathcal{W}_{\xi_\pi}=\max_{\|\Gamma_\xi\|\leq 1}\mathbb{E}_{s,a\sim\xi_\pi}\mathbb{E}_{\bar{a}\sim\phi^{\mathcal{A}}(\,\cdot\mid\phi(s),a)}\mathbb{E}_{\bar{s}'\sim\overline{\mathbf{P}}(\,\cdot\mid\bar{s},\bar{a})}\,\Gamma_\xi\left(\phi\left(s\right),\bar{a},\bar{s}'\right)-\mathbb{E}_{\bar{s},\bar{a},\bar{s}'\sim\xi_{\bar{\pi}}}\Gamma_\xi\left(\bar{s},\bar{a},\bar{s}'\right)$
- $L_{\mathbf{P}}^{\xi_\pi}=\max_{\|\Gamma_{\mathbf{P}}\|\leq 1}\mathbb{E}_{s,a,s'\sim\xi_\pi}\mathbb{E}_{\bar{s},\bar{a},\bar{s}'\sim\phi(\,\cdot\mid s,a,s')}\left[\Gamma_{\mathbf{P}}\left(s,a,\bar{s},\bar{a},\bar{s}'\right)-\mathbb{E}_{\bar{s}^\star\sim\overline{\mathbf{P}}(\,\cdot\mid\bar{s},\bar{a})}\Gamma_{\mathbf{P}}\left(s,a,\bar{s},\bar{a},\bar{s}^\star\right)\right]$

Wasserstein Auto-encoded Markov Decision Process

$$\min \mathbb{E}_{s,a,s'\sim\xi_\pi}\mathbb{E}_{\bar{s},\bar{a},\bar{s}'\sim\phi(\,\cdot\mid s,a,s')}\left[d_{\mathcal{S}}\left(s,\varsigma\left(\bar{s}\right)\right)+d_{\mathcal{A}}\left(a,\psi\left(\bar{s},\bar{a}\right)\right)+d_{\mathcal{S}}\left(s',\varsigma\left(\bar{s}'\right)\right)\right]+L_{\mathcal{R}}^{\xi_\pi}+\beta\left(\mathcal{W}_{\xi_\pi}+L_{\mathbf{P}}^{\xi_\pi}\right)$$



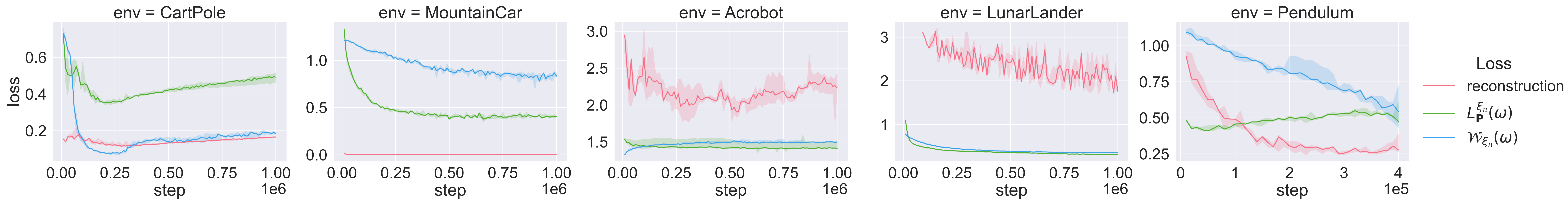
Mini-max learning procedure

- $\mathcal{W}_{\xi_\pi} = \max_{\|\Gamma_\xi\|\leq 1} \mathbb{E}_{s,a\sim\xi_\pi}\mathbb{E}_{\bar{a}\sim\phi^{\mathcal{A}}(\,\cdot\mid\phi(s),a)}\mathbb{E}_{\bar{s}'\sim\bar{\mathbf{P}}(\,\cdot\mid\bar{s},\bar{a})}\Gamma_\xi\left(\phi\left(s\right),\bar{a},\bar{s}'\right)-\mathbb{E}_{\bar{s},\bar{a},\bar{s}'\sim\xi_{\bar{\pi}}}\Gamma_\xi\left(\bar{s},\bar{a},\bar{s}'\right)$
- $L_{\mathbf{P}}^{\xi_\pi} = \max_{\|\Gamma_{\mathbf{P}}\|\leq 1} \mathbb{E}_{s,a,s'\sim\xi_\pi}\mathbb{E}_{\bar{s},\bar{a},\bar{s}'\sim\phi(\,\cdot\mid s,a,s')}\left[\Gamma_{\mathbf{P}}\left(s,a,\bar{s},\bar{a},\bar{s}'\right)-\mathbb{E}_{\bar{s}^\star\sim\bar{\mathbf{P}}(\,\cdot\mid\bar{s},\bar{a})}\Gamma_{\mathbf{P}}\left(s,a,\bar{s},\bar{a},\bar{s}^\star\right)\right]$

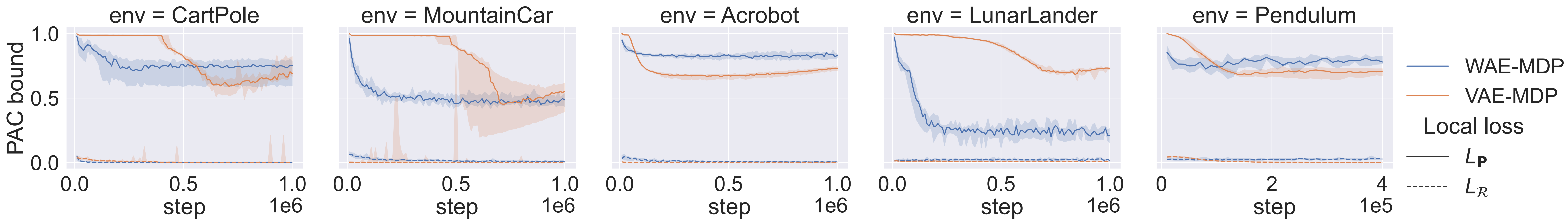
Discriminators distinguish between latent variables that can be generated from the latent MDP and those that cannot.

Evaluation

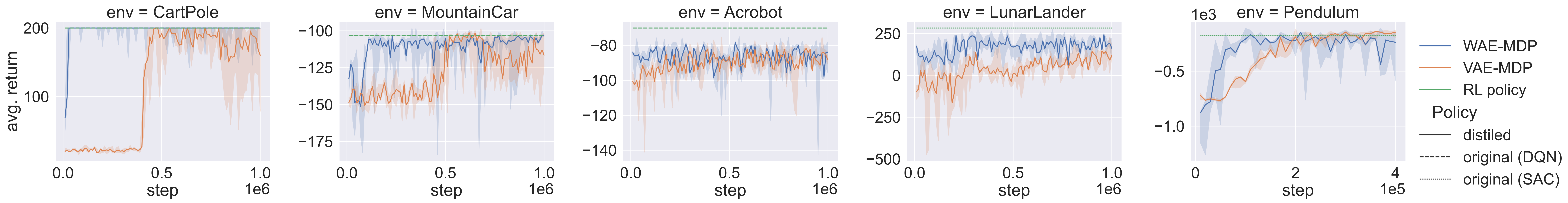
WAE-MDP Losses (Reconstruction Loss + Regularizers)



Local Losses (PAC evaluation)

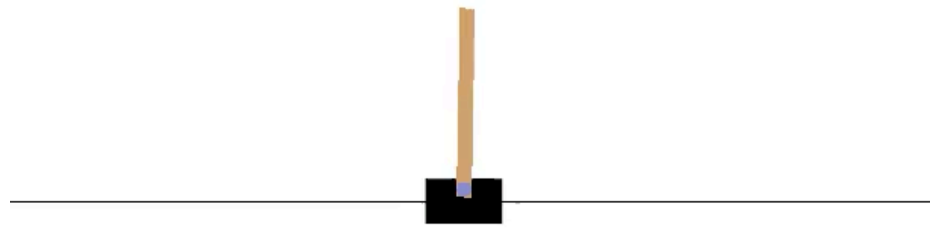


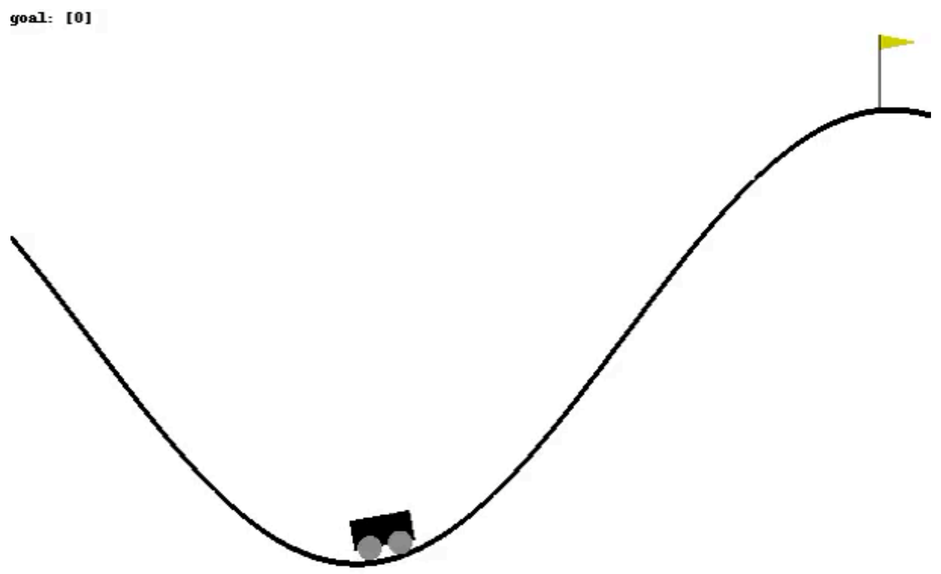
Distillation: performance of $\bar{\pi}$

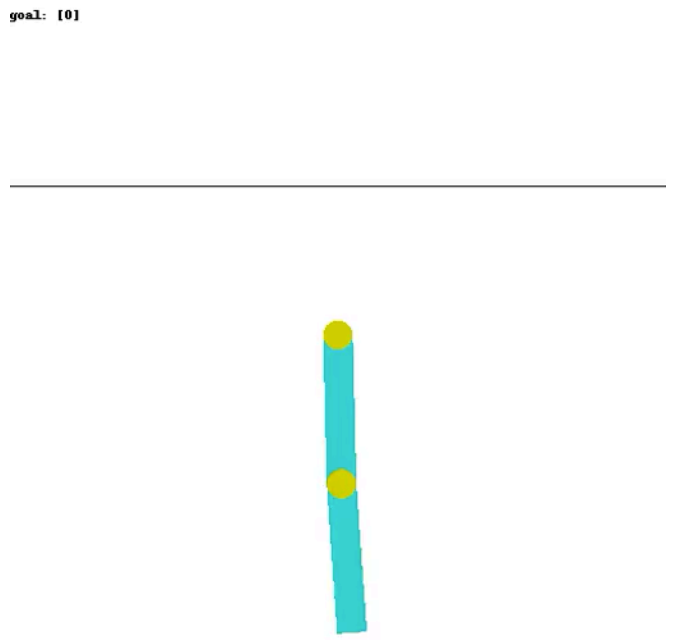


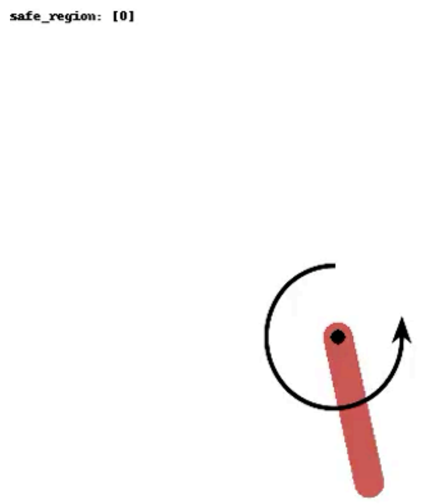
Evaluation

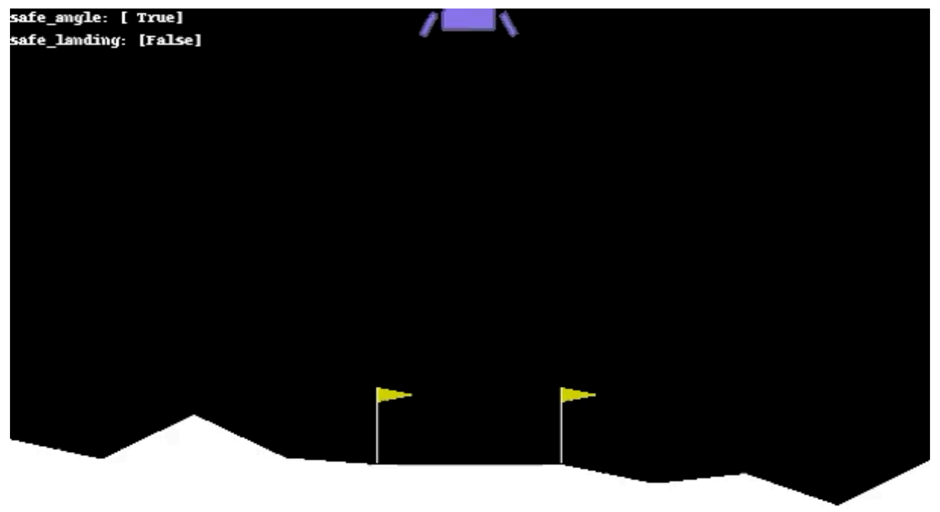
CartPole *MountainCar* *Acrobot* *Pendulum* *LunarLander*











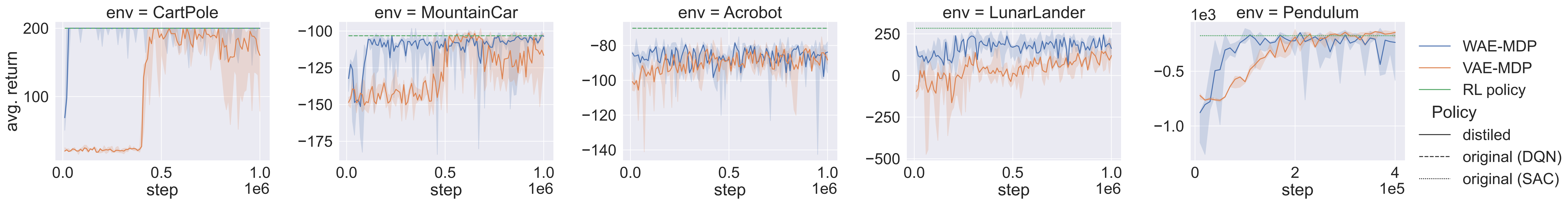
Time-to-failure properties (lower is better)

$\varphi = \neg\text{Reset}\ \mathcal{U}\ \neg\text{Safe}$ $\varphi = \neg\text{Goal}\ \mathcal{U}\ \text{Reset}$ $\varphi = \neg\text{Goal}\ \mathcal{U}\ \text{Reset}$ $\varphi = \Diamond(\neg\text{Safe} \wedge \bigcirc\text{Reset})$ $\varphi = \neg\text{SafeLanding}\ \mathcal{U}\ \text{Reset}$

$\overline{V}^{\varphi}_{\bar{\pi}_\theta}(\bar{s}_I) = 0.032$ $\overline{V}^{\varphi}_{\bar{\pi}_\theta}(\bar{s}_I) = 0$ $\overline{V}^{\varphi}_{\bar{\pi}_\theta}(\bar{s}_I) = 0.0022$ $\overline{V}^{\varphi}_{\bar{\pi}_\theta}(\bar{s}_I) = 0.037$ $\overline{V}^{\varphi}_{\bar{\pi}_\theta}(\bar{s}_I) = 0.0702$

Conclusion

WAE-MDPs distill original RL policies up to 10 times faster than VAE-MDPs



- **(V-, W)*AE-MDPs***, frameworks for learning **discrete latent models** of **unknown continuous-spaces** environment with **bisimulation guarantees**
  - **Enable** the **verification** of **Deep RL policies** by *distilling* the agent behaviours over a **tractable, simpler, bisimilar latent space model**
  - The **guarantees** obtained by **model checking** the distilled policy in the latent model can be *lifted* to the real environment thanks to the **bisimulation guarantees**
  - WAE-MDPs overcome the limits of VAEs by directly incorporating bisimulation metrics in its optimisation function

Further Work: Beyond Distillation

Application to POMDPs

Published as a conference paper at ICLR 2024

THE WASSERSTEIN BELIEVER
LEARNING BELIEF UPDATES FOR PARTIALLY OBSERVABLE ENVIRONMENTS THROUGH RELIABLE LATENT SPACE MODELS

Raphael Avalos[1 *] **Florent Delgrange**[1,2 *]
Ann Nowé[1 †] **Guillermo A. Pérez**[2,3 †] **Diederik M. Roijers**[1,4 †]
[1] AI Lab, Vrije Universiteit Brussel (Belgium) [2] University of Antwerp (Belgium)
[3] Flanders Make (Belgium) [4] City of Amsterdam (The Netherlands)
{raphael.avalos, florent.delgrange}@vub.be

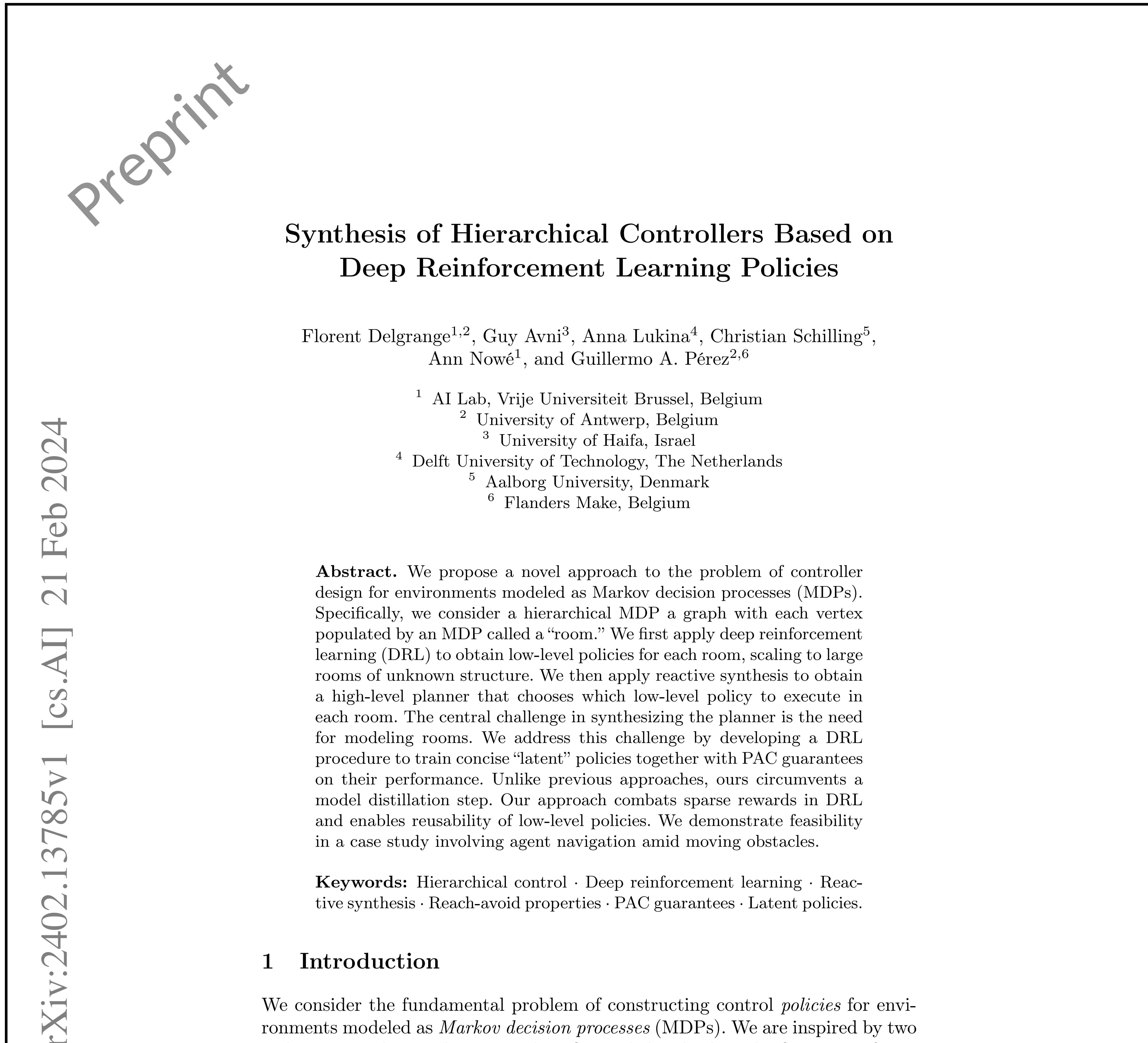
ABSTRACT

Partially Observable Markov Decision Processes (POMDPs) are used to model environments where the state cannot be perceived, necessitating reasoning based on past observations and actions. However, remembering the full history is generally intractable due to the exponential growth in the history space. Maintaining a probability distribution that models the belief over the current state can be used as a sufficient statistic of the history, but its computation requires access to the model of the environment and is often intractable. While SOTA algorithms use Recurrent Neural Networks to compress the observation-action history aiming to learn a sufficient statistic, they lack guarantees of success and can lead to sub-optimal policies. To overcome this, we propose the Wasserstein Belief Updater, an RL algorithm that learns a latent model of the POMDP and an approximation of the belief update under the assumption that the state is observable during training. Our approach comes with theoretical guarantees on the quality of our approximation ensuring that our latent beliefs allow for learning the optimal value function.

1 INTRODUCTION

Partially Observable Markov Decision Processes (POMDPs) define a powerful framework for modeling decision-making in uncertain environments where the state is not fully observable. These problems are common in many real-world applications, such as robotics (Lauri et al., 2023), and recommendation systems (Wu et al., 2021). In contrast to *Markov Decision Processes* (MDPs), in a POMDP the agent perceives an imperfect observation of the state that does not suffice as conditioning signal for an optimal policy. As such, optimal policies must take the entire interaction history into account. As the space of possible histories scales exponentially in the length of the episode, using histories to condition policies is generally intractable. An alternative is the notion of *belief*,

Synthesis from RL components



Preprint

arXiv:2402.13785v1 [cs.AI] 21 Feb 2024

Synthesis of Hierarchical Controllers Based on Deep Reinforcement Learning Policies

Florent Delgrange[1,2], Guy Avni[3], Anna Lukina[4], Christian Schilling[5], Ann Nowé[1], and Guillermo A. Pérez[2,6]

[1] AI Lab, Vrije Universiteit Brussel, Belgium
[2] University of Antwerp, Belgium
[3] University of Haifa, Israel
[4] Delft University of Technology, The Netherlands
[5] Aalborg University, Denmark
[6] Flanders Make, Belgium

Abstract. We propose a novel approach to the problem of controller design for environments modeled as Markov decision processes (MDPs). Specifically, we consider a hierarchical MDP a graph with each vertex populated by an MDP called a "room." We first apply deep reinforcement learning (DRL) to obtain low-level policies for each room, scaling to large rooms of unknown structure. We then apply reactive synthesis to obtain a high-level planner that chooses which low-level policy to execute in each room. The central challenge in synthesizing the planner is the need for modeling rooms. We address this challenge by developing a DRL procedure to train concise "latent" policies together with PAC guarantees on their performance. Unlike previous approaches, ours circumvents a model distillation step. Our approach combats sparse rewards in DRL and enables reusability of low-level policies. We demonstrate feasibility in a case study involving agent navigation amid moving obstacles.

Keywords: Hierarchical control · Deep reinforcement learning · Reactive synthesis · Reach-avoid properties · PAC guarantees · Latent policies.

1 Introduction

We consider the fundamental problem of constructing control *policies* for environments modeled as *Markov decision processes* (MDPs). We are inspired by two