Distillation of RL Policies through Bisimilar Latent Models with Formal Guarantees

Florent Delgrange, Ann Nowé, Guillermo A. Pérez
Reinforcement Learning

- Unknown environment
- Continuous state/action spaces
Overview

Reinforcement Learning

- Unknown environment
- Continuous state/action spaces
Overview

Reinforcement Learning Policies with Formal Guarantees

- Unknown environment
- Continuous state/action spaces

Model Checking

- Formal guarantees
  - 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** $\mathcal{S}$
- **Action space** $\mathcal{A}$
- **Reward function** $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$
- **Probability transition function** $P(s' | s, a)$
- **Atomic propositions** $\mathcal{AP}$ and labelling function $\ell : \mathcal{S} \rightarrow 2^{\mathcal{AP}}$

**Policies** prescribe which action to choose at each step: $\pi : \mathcal{S} \rightarrow \Delta(\mathcal{A})$, $a_t \sim \pi(\cdot | s_t)$

**Value functions:**

1. **Discounted return:** $V_\pi(s) = \mathbb{E}_\pi \left[ \sum_{t=0}^{\infty} \gamma^t \cdot \mathcal{R}(s_t, a_t) \mid s_0 = s \right]$

2. **Properties** $\varphi$: $\lim_{\gamma \rightarrow \infty} V_\pi(s, \varphi) = P_\pi(s \models \varphi)$; e.g., $P_\pi(s \models \text{the agent reaches the goal})$
Bisimulation

Continuous-spaces MDP

\[ \mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \ell \rangle \]

Discrete latent MDP

\[ \overline{\mathcal{M}} = \langle \overline{\mathcal{S}}, \overline{\mathcal{A}}, \overline{\mathcal{R}}, \overline{\mathcal{P}}, \ell \rangle \]

state embedding
\[ \phi(s) = \overline{s} \]

action embedding
\[ \psi(\overline{s}, \overline{a}) = a \]
Bisimulation

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

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

**Largest:** ~

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

$$M = (S, A, \mathcal{R}, P, \ell)$$

**Discrete latent MDP**

$$\overline{M} = (\overline{S}, \overline{A}, \overline{\mathcal{R}}, \overline{P}, \overline{\ell})$$
Bisimulation

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

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

Largest: $\sim$

- 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.,

$\rightarrow$ They behave the same

$\circ$ All or nothing: two states nearly identical with slight numerical difference $\epsilon$ are $\neq$

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

- For policy $\pi$, $\gamma \in [0,1]$, and formal logic $\mathcal{L}$:
  - **Bisimulation distance:** largest behavioral difference (de Alfaro et. al, 2003; Desharnais et. al, 2004)
    \[ \tilde{d}_\pi(s_1, s_2) = \sup_{\phi \in \mathcal{L}_\gamma} \left| V_\pi(s_1, \phi) - V_\pi(s_2, \phi) \right| \quad \forall s_1, s_2 \in \mathcal{S} \]
    Take the values of the {event / specification / property} leading to the largest difference
  - **Kernel is bisimilarity:** $\tilde{d}_\pi(s_1, s_2) = 0 \iff s_1 \sim s_2$
Execution of a latent policy $\tilde{\pi}$ in the original model: Local Losses

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

\[
L_{\xi_{\pi}}^{\tilde{\pi}} = \mathbb{E}_{s,a \sim \xi_{\pi}} W_{d_{\xi}} \left( \phi P \left( \cdot \mid s, a \right), \tilde{P} \left( \cdot \mid \phi(s), \tilde{a} \right) \right)
\]

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

- Abstraction quality: for all $s_1, s_2 \in \mathcal{S}$ such that $\phi(s_1) = \phi(s_2)$

\[
\tilde{d}_{\tilde{\pi}} (s_1, s_2) \leq \left( \frac{L_{\xi_{\pi}}^{\tilde{\pi}} + \gamma L_{\xi_{\pi}}^{\tilde{\pi}}}{1 - \gamma} \right) \cdot \left( \frac{\xi_{\pi}^{-1} (s_1) + \xi_{\pi}^{-1} (s_2)}{\xi_{\pi}^{-1} (s_1) + \xi_{\pi}^{-1} (s_2)} \right)
\]
**Execution of a latent policy \( \bar{\pi} \) in the original model:** Local Losses

- Latent policy \( \pi \), stationary distribution \( \xi \):
  
  \[
  L_{\xi_p}^\pi = \mathbb{E}_{s,a \sim \xi} W_{d_\pi} \left( \phi P \left( \cdot | s, \bar{a} \right), \bar{P} \left( \cdot | \phi(s), \bar{a} \right) \right)
  \]
  \[
  L_{\xi_\mathcal{R}}^\pi = \mathbb{E}_{s,a \sim \xi} \left| \mathcal{R} (s, \bar{a}) - \bar{\mathcal{R}} (\phi(s), \bar{a}) \right|
  \]

- **Abstraction quality:**
  
  \[
  \mathbb{E}_{s \sim \xi} \left| V_{\bar{\mathcal{R}}}(s) - \bar{V}_{\pi}(s) \right| \leq \frac{L_{\xi_\mathcal{R}}^\pi + \gamma L_{\xi_p}^\pi}{1 - \gamma}
  \]

- **Representation quality:** for all \( s_1, s_2 \in \mathcal{S} \) such that \( \phi(s_1) = \phi(s_2) \)

  \[
  \left| V_{\bar{\mathcal{R}}}(s_1) - V_{\pi}(s_2) \right| \leq \left( \frac{L_{\xi_\mathcal{R}}^\pi + \gamma L_{\xi_p}^\pi}{1 - \gamma} \right) \cdot \left( \xi_{\bar{\pi}}^{-1}(s_1) + \xi_{\bar{\pi}}^{-1}(s_2) \right)
  \]
Execution of a latent policy $\bar{\pi}$ in the original model: Local Losses

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

\[
L_{P}^{\bar{\xi}_\pi} = \mathbb{E}_{s, \bar{a} \sim \bar{\xi}_\pi} W_{d_\bar{\pi}} \left( \phi P \left( \cdot | s, \bar{a} \right), \bar{P} \left( \cdot | \phi(s), \bar{a} \right) \right)
\]

\[
L_{R}^{\bar{\xi}_\pi} = \mathbb{E}_{s, \bar{a} \sim \bar{\xi}_\pi} \left| R(s, \bar{a}) - \bar{R}(\phi(s), \bar{a}) \right|
\]

- Abstraction quality: $\mathbb{E}_{s \sim \bar{\xi}_\pi} \left| V_{\bar{\pi}}(s) - \bar{V}_{\pi}(s) \right| \leq \frac{L_{R}^{\bar{\xi}_\pi} + \gamma L_{P}^{\bar{\xi}_\pi}}{1 - \gamma}$

- Representation quality: for all $s_1, s_2 \in S'$ such that $\phi(s_1) = \phi(s_2)$

\[
\left| V_{\bar{\pi}}(s_1) - V_{\pi}(s_2) \right| \leq \left( \frac{L_{R}^{\bar{\xi}_\pi} + \gamma L_{P}^{\bar{\xi}_\pi}}{1 - \gamma} \right) \cdot \left( \frac{\bar{\xi}_\pi^{-1}(s_1) + \bar{\xi}_\pi^{-1}(s_2)}{2} \right)
\]

- PAC scheme from samples: let trace $\langle s_0; T, \bar{a}_{0; T-1}, r_0; T-1 \rangle \sim \bar{\xi}_\pi$, $\epsilon, \delta \in [0, 1]$ and $T \geq \left\lceil \frac{-\log \left( \delta/4 \right)}{2\epsilon^2} \right\rceil$:

\[
\hat{L}_{R}^{\bar{\xi}_\pi} = \frac{1}{T} \sum_{t=0}^{T-1} \left| r_t - \bar{R}(\phi(s_t), \bar{a}_t) \right| \quad \text{and} \quad \hat{L}_{P}^{\bar{\xi}_\pi} = \frac{1}{T} \sum_{t=0}^{T-1} \left[ 1 - \bar{P}(\phi(s_{t+1}) | \phi(s_t), \bar{a}_t) \right]
\]

Then, $\left| L_{R}^{\bar{\xi}_\pi} - \hat{L}_{R}^{\bar{\xi}_\pi} \right| \leq \epsilon$ and $\left| L_{P}^{\bar{\xi}_\pi} - \hat{L}_{P}^{\bar{\xi}_\pi} \right| \leq \epsilon$ with probability $1 - \delta$
Learning the Latent Space Model

- Train a *behavioral model* $\bar{\xi}_\theta$ by learning from traces produced by executing the RL policy $\pi$ in the original model $\mathcal{M}$

- **Goal**: learn $\xi_\theta$ so that we can retrieve:
  - The latent MDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, P, \ell \rangle$
  - The embedding functions $\phi, \psi$
  - A latent policy $\bar{\pi}$ distilled from $\pi$

- Minimize a *discrepancy* $D$ between $\mathcal{M} \otimes \pi$ and $\bar{\xi}_\theta$

$$
\min_{\theta} D_{KL} (\mathcal{M} \otimes \pi, \bar{\xi}_\theta)
$$

- **Choose the Kullback-Leibler divergence**

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

- Train a behavioral model $\xi_\theta$ by learning from traces produced by executing the RL policy $\pi$ in the original model $M$.

- **Goal**: learn $\xi_\theta$ so that we can retrieve:
  - The latent MDP $\overline{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \ell \rangle$
  - The embedding functions $\phi, \psi$
  - A latent policy $\bar{\pi}$ distilled from $\pi$

- Minimize a discrepancy $D$ between $M \otimes \pi$ and $\xi_\theta$

$$
\min_\theta D_{KL}(M \otimes \pi, \xi_\theta) \equiv \max_\theta \mathbb{E}_{\tau \sim \overline{M} \otimes \pi} \left[ \log \xi_\theta(\tau) \right] \geq \max_{i, \theta} \text{ELBO}(\overline{M}_\theta, \phi_i, \psi_\theta)
$$

- Choose the Kullback-Leibler divergence

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

(Kingma & Welling, 2014; Hoffman et al., 2013)
Variational Markov Decision Process

\[
\max_{\lambda, \theta} ELBO \left( \mathcal{M}_\theta, \phi_\lambda, \psi_\theta \right) = - \min_{\lambda, \theta} \left\{ D_{\lambda, \theta} + R_{\lambda, \theta} \right\}
\]
Learning the environment abstraction through

\[ \text{max} \quad ELBO \left( \mathcal{M}_\theta, \phi_\theta, \psi_\theta \right) = - \min \{ D_{t,\theta} + R_{t,\theta} \} \]

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

\[ D_{t,\theta} = - \mathbb{E}_{s,a,r,s',\sim E_n} \left[ \log P_{t,\theta}^s (s' | s') + \log \psi_\theta (a | s, s') + \log P_{t,\theta}^r (r | s, a) \right] \]

\[ R_{t,\theta} = \mathbb{E}_{s,a,s',\sim E_n} \left[ D_{KL} \left( \phi_t (\cdot | s') \parallel \tilde{P}_\theta (\cdot | s, a) \right) + D_{KL} \left( Q_t^A (\cdot | s, a) \parallel \tilde{\pi}_\theta (\cdot | s) \right) \right] \]

\[ L_{\psi} = \mathbb{E}_{s,\sim \xi_\psi} W_{\psi} \left( \phi (\cdot | s,a), \tilde{P} (\cdot | \phi(s), a) \right) \leq \mathbb{E}_{s,\sim \xi_\psi} W_{\psi} \left( \phi (\cdot | s'), \tilde{P} (\cdot | \phi(s), a) \right) \]
Variational Markov Decision Process

\[
\max_{\mathcal{M}_\theta, \phi, \psi_\theta} \text{ELBO}(\mathcal{M}_\theta, \phi, \psi_\theta) = - \min_{D_{t,\theta} + R_{t,\theta}}
\]

\[
D_{t,\theta} = - \mathbb{E}_{s, a, r, s' \sim E_n} \left[ \log P^r_\theta(s' | \bar{s}') + \log P^\psi_\theta(a | \bar{s}, \bar{a}) + \log P^R_\theta(r | \bar{s}, \bar{a}) \right]
\]

\[
R_{t,\theta} = \mathbb{E}_{s, a, s' \sim E_n} \left[ D_{KL}(\phi(. | s') || \bar{P}_\theta(. | \bar{s}, \bar{a})) + D_{KL}(Q^A_{\pi}(. | \bar{s}, a) || \bar{P}_\theta(. | \bar{s})) \right]
\]

\[
L_{\psi} = \mathbb{E}_{s, \bar{s} \sim E_n} W_{d_\mathcal{E}} \left( \phi P(\cdot | s, \bar{a}) \bar{P}(\cdot | \phi(s), \bar{a}) \right)
\leq \mathbb{E}_{s, \pi, \bar{s} \sim E_n} W_{d_\mathcal{E}} \left( \phi P(\cdot | s', \bar{a}) \bar{P}(\cdot | \phi(s), \bar{a}) \right)
\]

- 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
**Distillation: performance of \( \bar{\pi} \)**

*Handling posterior collapse slows down the learning process*
Learning the Latent Space Model

- Train a behavioral model $\xi_\theta$ by learning from traces produced by executing the RL policy $\pi$ in the original model $M$.

- **Goal**: learn $\xi_\theta$ so that we can retrieve:
  - The latent MDP $\overline{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathbb{P}, \ell \rangle$
  - The embedding functions $\phi, \psi$
  - A latent policy $\tilde{\pi}$ distilled from $\pi$

- Minimize a discrepancy $D$ between $\overline{M} \otimes \pi$ and $\xi_\theta$

\[
\min_\theta W \left( \overline{M} \otimes \pi, \xi_\theta \right)
\]

- Choose the Wasserstein Distance

\[
W \left( P, Q \right) = \inf_{\lambda \in \Lambda(P, Q)} \mathbb{E}_{x, y \sim \lambda} d \left( x, y \right) = \sup_{\|f\| \leq 1} \mathbb{E}_{x \sim P} f(x) - \mathbb{E}_{y \sim Q} f(y)
\]
Learning the Latent Space Model

- Train a \textit{behavioral model} $\xi_\theta$ by learning from traces produced by executing the RL policy $\pi$ in the original model $\mathcal{M}$

- \textbf{Goal:} learn $\xi_\theta$ so that we can retrieve:
  - The latent MDP $\overline{\mathcal{M}} = \langle \overline{\mathcal{S}}, \overline{\mathcal{A}}, \overline{\mathcal{R}}, \overline{\mathbb{P}}, \ell' \rangle$
  - The embedding functions $\phi, \psi$
  - A latent policy $\bar{\pi}$ distilled from $\pi$

- Minimize a \textit{discrepancy} $D$ between $\mathcal{M} \otimes \pi$ and $\xi_\theta$

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

- Choose the \textbf{Wasserstein Distance}

$$
W(P, Q) = \inf_{\lambda \in \Lambda(P, Q)} \mathbb{E}_{x,y \sim \lambda} d(x, y) = \sup_{\|f\|_1 \leq 1} \mathbb{E}_{x \sim P} f(x) - \mathbb{E}_{y \sim Q} f(y)
$$
Wasserstein Auto-encoded Markov Decision Process

\[
\min \mathbb{E}_{s,a,s' \sim \xi} \mathbb{E}_{\tilde{s}, \tilde{a}, \tilde{s}' \sim \phi} \left[ d_\mathcal{S}(s, \zeta(\tilde{s})) + d_\mathcal{A}(a, \psi(\tilde{s}, \tilde{a})) + d_\mathcal{S}(s', \zeta(\tilde{s}')) \right] + L_\xi^\pi + \beta \left( \mathcal{W}_\xi^\pi + L_\xi^\pi \right)
\]

- \( \mathcal{W}_\xi^\pi = \max_{\| \Gamma_{\zeta} \| \leq 1} \mathbb{E}_{s,a \sim \xi} \mathbb{E}_{\tilde{a} \sim \phi} (\cdot | \phi(s), a) \mathbb{E}_{\tilde{s} \sim \mathcal{P}} (\cdot | \tilde{s}, \tilde{a}) \Gamma_{\zeta} \left( \phi(s), \tilde{a}, \tilde{s} \right) - \mathbb{E}_{\tilde{s}, \tilde{a}, \tilde{s}' \sim \xi} \Gamma_{\zeta} \left( \tilde{s}, \tilde{a}, \tilde{s}' \right) \)

- \( L_\pi^\mathcal{R} = \sum_{\mathcal{R}} (\cdot | \cdot) \)
Wasserstein Auto-encoded Markov Decision Process

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

Mini-max learning procedure

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 $\tilde{\pi}$
Evaluation

CartPole  MountainCar  Acrobot  Pendulum  LunarLander

Time-to-failure properties (lower is better)

\[ \varphi = \neg \text{Reset } \mathcal{U} \neg \text{Safe} \]
\[ \varphi = \neg \text{Goal } \mathcal{U} \text{ Reset} \]
\[ \varphi = \neg \text{Goal } \mathcal{U} \text{ Reset} \]
\[ \varphi = \Diamond (\neg \text{Safe } \land \Box \text{Reset}) \]
\[ \varphi = \neg \text{Safe Landing } \mathcal{U} \text{ Reset} \]

\[ \mathbb{V}_{\pi_0}^\varphi (\bar{s}_t) = 0.032 \]
\[ \mathbb{V}_{\pi_0}^\varphi (\bar{s}_t) = 0 \]
\[ \mathbb{V}_{\pi_0}^\varphi (\bar{s}_t) = 0.0022 \]
\[ \mathbb{V}_{\pi_0}^\varphi (\bar{s}_t) = 0.037 \]
\[ \mathbb{V}_{\pi_0}^\varphi (\bar{s}_t) = 0.0702 \]
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 Avalos1,2, Florent Delgrange1,2, 6
Ann Nowé1,2, Guillermo A. Pérez3,6, Diederik M. Roijers1,2, 6
1 AI Lab, Vrije Universiteit Brussel (Belgium) 2 University of Antwerp (Belgium)
3 Flanders Make (Belgium) 6 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, maintaining a sufficient statistic of the history (Kaelbling et al., 1998) but the computation of their closed-form expression require the access to a model of the environment and is in general intractable.

To overcome those challenges, SOTA algorithms compress the history into a fixed-size vector with a sufficient statistic, RNNs can be combined with regularization techniques, including generative models (Chen et al., 2022; Hafner et al., 2019; 2021), particle filtering (Igl et al., 2018; Ma et al., 2020), and predicting distant observations (Gregor et al., 2018; 2019). However, none of these techniques with complementary benefits and drawbacks. The first is reinforcement learning (RL) [52], where the designer chooses how rewards are issued for 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 combines 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