Distillation of RL Policies with Formal Guarantees via Variational Abstraction of Markov Decision Processes

Florent Delgrange,[1] **Ann Nowé,**[1] **Guillermo A. Pérez**[2]

[1] AI Lab, Vrije Universiteit Brussel
florent.delgrange@ai.vub.ac.be
[2] University of Antwerp – Flanders Make

Abstract

We consider the challenge of policy simplification and verification in the context of policies learned through reinforcement learning (RL) in continuous environments. In well-behaved settings, RL algorithms have convergence guarantees in the limit. While these guarantees are valuable, they are insufficient for safety-critical applications. Furthermore, they are lost when applying advanced techniques such as deep-RL. To recover guarantees when applying advanced RL algorithms to more complex environments with (i) reachability, (ii) safety-constrained reachability, or (iii) discounted-reward objectives, we build upon the DeepMDP framework introduced by Gelada et al. to derive new bisimulation bounds between the unknown environment and a learned discrete latent model of it. Our bisimulation bounds enable the application of formal methods for Markov decision processes. Finally, we show how one can use a policy obtained via state-of-the-art RL to efficiently train a variational autoencoder that yields a discrete latent model with provably approximately correct bisimulation guarantees. Additionally, we obtain a distilled version of the policy for the latent model.

1 Introduction

While *reinforcement learning* (RL) has been applied to a wide range of challenging domains, from game playing (Mnih et al. 2015) to real-world applications such as effective canal control (Ren et al. 2021), more widespread deployment in the real world is hampered by the lack of guarantees provided with the learned policies. Although there are RL algorithms which have limit-convergence guarantees in the discrete setting (Tsitsiklis 1994) — and even in some continuous settings with function approximation, e.g., Nowe (1994) — these are lost when applying more advanced techniques which make use of general nonlinear function approximators (Tsitsiklis and Roy 1997) to deal with continuous *Markov decision processes* (MDPs) such as *deep*-RL (e.g., Mnih et al. 2015). In this paper, we apply such advanced RL algorithms to unknown continuous MDPs with (i) reachability, (ii) safety-constrained reachability, or (iii) discounted-reward objectives. To recover the formal guarantees, we use the obtained policy to train a *variational autoencoder* (VAE) which gives us a *discrete latent model* that approximates the unknown environment. We build upon the *DeepMDP* framework (Gelada et al. 2019) to provide guarantees on the quality of the abstraction induced by this model. DeepMDPs are provided with such guarantees when their *loss functions* are minimized. These can be defined on the entire state space (*global*) or on states visited under a given policy (*local*). The guarantees concern a *state embedding function*, linking the latent and original MDPs and are defined as bounds on the difference of their *value function* and *bisimulation distance*. The latter was only developed for global losses. While these are interesting in theory, they are often infeasible to measure in practice. In contrast, we introduce such bounds in the local setting and further consider an *action embedding function* to handle continuous actions. Importantly, we focus on general MDPs and do not restrict our attention to deterministic ones as was done by Gelada et al. to enable the approximation and minimization of their losses via neural networks. We also give PAC approximation schemes to compute both the losses and said bounds.

Our VAE is trained by maximizing a lower bound on the likelihood of traces generated by executing the RL policy in the environment. We derive a loss function, incorporating variational versions of the local losses, that enables learning (i) discrete state and action spaces, (ii) an MDP defined over these spaces, (iii) state and action embedding functions, linking the original and discrete MDPs, and (iv) a *distilled* version of the RL policy which can be executed in both models using the learned latent spaces. An important challenge for our approach is the *posterior collapse problem* which often occurs when optimizing a variational model (e.g., Alemi et al. 2018). We present a novel approach based on *prioritized experience replay* (Schaul et al. 2016) to resolve this when learning a discrete latent model.

All of the above result in an efficient way of training a VAE to obtain a discrete latent model that is provably approximately bisimilar to the unknown MDP, further yielding a distilled version of the RL policy. These enable the application of formal methods and tools that have been developed for discrete MDPs: for instance, PRISM (Kwiatkowska, Norman, and Parker 2011), MODEST (Hartmanns and Hermanns 2014), and STORM (Hensel et al. 2021).

Other related work. Frameworks providing formal guarantees *during the RL process* include the work of Junges et al. (2016), *Shielded-RL* (Alshiekh et al. 2018; Jansen et al.

2020), and *AlwaysSafe* (Simão, Jansen, and Spaan 2021). These all require an abstract model of the safety aspect of the environment. Our approach is complementary in that we assume no prior knowledge and *learn an abstraction*. Notably, our goal is not the same: they aim at verifying whether the exploration is safe while our goal is to verify policies learned via *any* RL technique. The MOSAIC approach (Bacci and Parker 2020) shares ours in the particular case of verifying deep-RL policies. However, they require (i) the neural network specifying the policy, (ii) the environment to be known and deterministic, and (iii) a formal description of the probability with which faults occur when attempting to execute particular actions. Finally, Carr, Jansen, and Topcu (2020) verify policies represented as recurrent neural networks (RNNs). Although they require the environment to be discrete as well as a formal model of the environment, the authors discretize the RNN hidden states by using *quantized* autoencoders, in the same spirit as our policy distillation.

VAEs have been used in the context of (model-based) RL to learn latent representations of the unknown environment and train simpler policies from the features extracted (e.g., Corneil, Gerstner, and Brea 2018; Freeman, Ha, and Metz 2019; Lee et al. 2020; Burden, Siahroudi, and Kudenko 2021). In particular, Corneil, Gerstner, and Brea (2018) focused on learning discrete latent MDPs from continuous-state environments with discrete actions (without guarantees nor distilled policies) to plan via *prioritized sweeping*.

2 Background

We write $[T] = \{n \in \mathbb{N} \mid n \leqslant T\}$. For $A \subseteq X$, we denote by $\mathbf{1}_A \colon X \to [1]$ the indicator function: $\mathbf{1}_A(a) = 1$ iff $a \in A$. Let $\mathcal{X}$ be a complete and separable space and $\Sigma(\mathcal{X})$ denote the set of all Borel subsets of $\mathcal{X}$. We write $\mathcal{P}(\mathcal{X})$ for the set of measures P defined on $\mathcal{X}$ and $Supp(P) = \{x \in \mathcal{X} \mid P(x) > 0\}$ to denote their support.

Discrepancy measures. Let $P, Q \in \mathcal{P}(\mathcal{X})$ with density functions p and q. Their discrepancy can be measured via

- *Kullback-Leibler* (KL) divergence: $D_{\mathrm{KL}}(P \parallel Q) = \mathbb{E}_{x\sim P}\big[\log(p(x)/q(x))\big]$.
- *Wasserstein*: $W_d(P, Q) = \inf_{\lambda\in\Lambda(P,Q)} \mathbb{E}_{x,y\sim\lambda}\, d(x, y)$, where $d \colon \mathcal{X} \to [0, \infty[$ is a distance metric over $\mathcal{X}$ and $\Lambda(P, Q)$ is the set of all *couplings* of P and Q.
- *Total Variation* (TV): $d_{TV}(P, Q) = \sup_{A\in\Sigma(\mathcal{X})} |P(A) - Q(A)|$. If $\mathcal{X}$ is equipped with the discrete metric $\mathbf{1}_{\neq}$, TV coincides with the Wasserstein measure.

Markov decision processes. A *Markov decision process* (MDP) is a tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathbf{P}, \mathcal{R}, \ell, \mathbf{AP}, s_I \rangle$ where $\mathcal{S}$ is a set of *states*; $\mathcal{A}$, a set of *actions*; $\mathbf{P} \colon \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$, a *probability transition function*; $\mathcal{R} \colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, a *reward function*; $\ell \colon \mathcal{S} \to 2^{\mathbf{AP}}$, a *labeling function* over a set of atomic propositions $\mathbf{AP}$; and $s_I \in \mathcal{S}$, the *initial state*. The set of *enabled actions* of $s \in \mathcal{S}$ is $Act(s) \subseteq \mathcal{A}$. We assume $Act(s) \neq \varnothing$ for all $s \in \mathcal{S}$. If $|Act(s)| = 1$ for all $s \in \mathcal{S}$, $\mathcal{M}$ is a fully stochastic process called a *Markov chain* (MC).

Let $\mathsf{T} \subseteq \mathbf{AP}$, we write $[\![\mathsf{T}]\!] = \{\, s \mid \ell(s) \cap \mathsf{T} \neq \varnothing \,\} \subseteq \mathcal{S}$ and $[\![\neg\mathsf{T}]\!] = \mathcal{S} \backslash [\![\mathsf{T}]\!]$. We assume $\mathbf{AP}$ and labels being respectively *one-hot* and binary encoded. We write $\mathcal{M}_s$ for the MDP obtained when we replace the initial state of $\mathcal{M}$ by $s \in \mathcal{S}$, $\mathcal{M} \oplus \mathcal{R}'$ when we replace the reward function by $\mathcal{R}'$, and $\mathcal{M}^{\circlearrowright B}$ when we make absorbing states from $B \subseteq \mathcal{S}$, i.e., by changing $\mathbf{P}(\cdot \mid s, a)$ to $\mathbf{P}'(\cdot \mid s, a)$ such that $\mathbf{P}'(B \mid s, a) = 1$ for all $s \in B$, $a \in Act(s)$. We refer to MDPs with continuous states or actions spaces as *continuous MDPs*. In that case, we assume $\mathcal{S}$ and $\mathcal{A}$ are complete separable metric spaces equipped with a Borel σ-algebra and $\ell^{-1}(\mathsf{T}) \in \Sigma(\mathcal{S})$ for any $\mathsf{T} \subseteq \mathbf{AP}$.

Trajectories. A *trajectory* τ of $\mathcal{M}$ is a sequence of states and actions $\tau = \langle s_{0:T}, a_{0:T-1} \rangle$ where $s_0 = s_I$, $s_{t+1} \sim \mathbf{P}(\cdot \mid s_t, a_t)$ and $a_t \in Act(s_t)$ for $t \in [T-1]$. The set of infinite trajectories of $\mathcal{M}$ is $Traj_{\mathcal{M}}$. An *execution trace* $\hat{\tau}$ of $\mathcal{M}$ is a trajectory that additionally records labels and rewards encountered. The set of execution traces of $\mathcal{M}$ is $Traces_{\mathcal{M}}$.

Policies. A *(memoryless) policy* $\pi \colon \mathcal{S} \to \mathcal{P}(\mathcal{A})$ of $\mathcal{M}$ is a stochastic mapping from states to actions such that $Supp(\pi(\cdot \mid s)) \subseteq Act(s)$. The set of memoryless policies of $\mathcal{M}$ is $\Pi^{\mathrm{ml}}_{\mathcal{M}}$. An MDP $\mathcal{M}$ and $\pi \in \Pi^{\mathrm{ml}}_{\mathcal{M}}$ induce an MC $\mathcal{M}_\pi$ along with a unique probability measure $\mathbb{P}^{\mathcal{M}}_\pi$ on the Borel σ-algebra over measurable subsets $E \subseteq Traj_{\mathcal{M}}$ (Puterman 1994). We drop the superscript when the context is clear. For $\pi \in \Pi^{\mathrm{ml}}_{\mathcal{M}}$, we denote by $\mathbf{P}_\pi \colon \mathcal{S} \to \mathcal{P}(\mathcal{S})$ the probability transition distribution of $\mathcal{M}_\pi$, and by $\mathcal{R}_\pi \colon \mathcal{S} \to \mathbb{R}$ its reward function. We write $\hat{\tau} = \langle s_{0:T}, a_{0:T-1}, r_{0:T-1}, l_{0:T} \rangle \sim \mathcal{M}_\pi$ for $\langle s_{0:T}, a_{0:T-1} \rangle \sim \mathbb{P}^{\mathcal{M}}_\pi$ with $\hat{\tau} \in Traces_{\mathcal{M}_\pi}$.

Stationary distributions. Let $\pi \in \Pi^{\mathrm{ml}}_{\mathcal{M}}$, $\xi^t_\pi : \mathcal{S} \to \mathcal{P}(\mathcal{S})$ with $\xi^t_\pi(s' \mid s) = \mathbb{P}^{\mathcal{M}_s}_\pi\big(\{\, s_{0:\infty}, a_{0:\infty} \mid s_t = s' \,\}\big)$ be the distribution giving the probability for the agent of being in each state of $\mathcal{M}_s$ after t steps, and $B \subseteq \mathcal{S}$. B is a *strongly connected component* (SCC) of $\mathcal{M}_\pi$ if for any pair of states $s, s' \in B$, $\xi^t_\pi(s' \mid s) > 0$ for some $t \in \mathbb{N}$. It is a *bottom SCC* (BSCC) if (i) B is a maximal SCC, and (ii) for each $s \in B$, $\mathbf{P}_\pi(B \mid s) = 1$. The unique stationary distribution of B is $\xi_\pi \in \mathcal{P}(B)$. We write $s, a \sim \xi_\pi$ as shorthand for first sampling s from ξ_π and then a from π. An MDP $\mathcal{M}$ is *ergodic* if for all $\pi \in \Pi^{\mathrm{ml}}_{\mathcal{M}}$, the state space of $\mathcal{M}_\pi$ consists of a unique aperiodic BSCC with $\xi_\pi = \lim_{t\to\infty} \xi^t_\pi(\cdot \mid s)$ for all $s \in \mathcal{S}$.

Events and functions. Let $\mathsf{C}, \mathsf{T} \subseteq \mathbf{AP}$, we define the *constrained reachability* (resp. *reachability*) event as $\mathsf{C}\,\mathcal{U}\,\mathsf{T} = \{\, s_{0:\infty}, a_{0:\infty} \mid \exists i \in \mathbb{N}, \forall j < i, s_j \in [\![\mathsf{C}]\!] \wedge s_i \in [\![\mathsf{T}]\!] \,\} \in \Sigma(Traj_{\mathcal{M}})$ (resp. $\Diamond\mathsf{T} = \neg\varnothing\,\mathcal{U}\,\mathsf{T}$). Safety w.r.t. a set of failure states T can be expressed as a safe-constrained reachability event to a safe destination C (resp. safety event) through $\neg\mathsf{T}\,\mathcal{U}\,\mathsf{C}$ (resp. $\square\neg\mathsf{T} = Traj_{\mathcal{M}} \backslash \Diamond\mathsf{T}$). Let $\gamma \in [0, 1[$, $\varphi \in \{\, \epsilon, \mathsf{C}\,\mathcal{U}\,\mathsf{T}, \Diamond\mathsf{T} \,\}$ where ϵ is the empty symbol, and $\mathcal{R}^{\mathsf{T}} = (1-\gamma)\mathbf{1}_{[\![\mathsf{T}]\!]\times\mathcal{A}}$, the *value* obtained by running $\pi \in \Pi^{\mathrm{ml}}_{\mathcal{M}}$ from state s in $\mathcal{M}$ is $V^\varphi_\pi(s) = \mathbb{E}^{\mathcal{M}[s,\varphi]}_\pi\big[\sum_{t=0}^\infty \gamma^t \mathcal{R}(s_t, a_t)\big]$. It corresponds to the *expected discounted* (i) *return* when $\varphi = \epsilon$ with $\mathcal{M}[s] = \mathcal{M}_s$, (ii) *constrained reachability* when $\varphi = \mathsf{C}\,\mathcal{U}\,\mathsf{T}$ with $\mathcal{M}[s, \mathsf{C}\,\mathcal{U}\,\mathsf{T}] = \mathcal{M}_s^{\circlearrowright [\![\neg\mathsf{C}]\!] \cup [\![\mathsf{T}]\!]} \oplus \mathcal{R}^{\mathsf{T}}$, (iii) *reachability* when $\varphi = \Diamond\mathsf{T}$ with $\mathcal{M}[s, \Diamond\mathsf{T}] = \mathcal{M}_s^{\circlearrowright [\![\mathsf{T}]\!]} \oplus \mathcal{R}^{\mathsf{T}}$. When $\varphi \in \{\, \mathsf{C}\,\mathcal{U}\,\mathsf{T}, \Diamond\mathsf{T} \,\}$, observe that $V^\varphi_\pi(t) = 1$ for $t \in [\![\mathsf{T}]\!]$ and $\lim_{\gamma\to 1} V^\varphi_\pi(s) = \mathbb{P}^{\mathcal{M}_s}_\pi(\varphi)$ for $s \in \mathcal{S}$. The *action-value function* is $Q^\varphi_\pi(s, a) = \mathcal{R}'(s, a) + \mathbb{E}_{s'\sim\mathbf{P}(\cdot|s,a)}\big[\gamma V^\varphi_\pi(s')\big]$, with $\mathcal{R}' = \mathcal{R}$ if $\varphi = \epsilon$ and $\mathcal{R}' = \mathcal{R}^{\mathsf{T}}$ otherwise.

3 Latent Space Models

Given the original (continuous, possibly unknown) environment modeled as an MDP, a *latent space model* is another (simpler, smaller, and explicit) MDP with state-action space linked to the original one via *embedding functions*. The latter can be optimized to minimize a *measure* between the two models. Formally, fix MDPs $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathbf{P}, \mathcal{R}, \ell, \mathbf{AP}, s_I \rangle$ and $\overline{\mathcal{M}} = \langle \overline{\mathcal{S}}, \overline{\mathcal{A}}, \overline{\mathbf{P}}, \overline{\mathcal{R}}, \bar{\ell}, \mathbf{AP}, \bar{s}_I \rangle$ such that $\overline{\mathcal{S}}$ is equipped with metric $d_{\overline{\mathcal{S}}}$. Let $\phi \colon \mathcal{S} \to \overline{\mathcal{S}}$ and $\psi \colon \mathcal{S} \times \overline{\mathcal{A}} \to \mathcal{A}$ be respectively state and action embedding functions. We refer to $\langle \overline{\mathcal{M}}, \phi, \psi \rangle$ as a latent space model of $\mathcal{M}$ and $\overline{\mathcal{M}}$ as its *latent MDP*. We write $\overline{\Pi}^{\mathrm{ml}} = \Pi^{\mathrm{ml}}_{\overline{\mathcal{M}}}$, and $\overline{Q}_{\bar{\pi}}$ for the action-value function of a policy $\bar{\pi} \in \overline{\Pi}^{\mathrm{ml}}$ in $\overline{\mathcal{M}}$. We also consider $\bar{\pi}$ as a policy in $\mathcal{M}$: states passed to $\bar{\pi}$ are embedded with ϕ, then actions executed are embedded with ψ. Let $\bar{\pi} \in \overline{\Pi}^{\mathrm{ml}}$ and $s \in \mathcal{S}$, we write $\bar{a} \sim \bar{\pi}(\cdot \mid s)$ for $\bar{a} \sim \bar{\pi}(\cdot \mid \phi(s))$ and $Q_{\bar{\pi}}(s, \bar{a})$ as shorthand for $Q_{\bar{\pi}}(s, \psi(s, \bar{a}))$.

A particular point of interest is to focus on discrete latent models, where $d_{\overline{\mathcal{S}}} = \mathbf{1}_{\neq}$. In the following, we adopt the latent space model formalism of Gelada et al. (2019).

Notations. Let $\bar{\pi} \in \overline{\Pi}^{\mathrm{ml}}$, we write $|\overline{\mathcal{R}}^{\star}_{\bar{\pi}}|$ for $\sup_{\bar{s} \in \overline{\mathcal{S}}} \left|\overline{\mathcal{R}}_{\bar{\pi}}(\bar{s})\right|$. We say that $\overline{\mathcal{M}}$ is $\langle K^{\bar{\pi}}_{\overline{\mathcal{R}}}, K^{\bar{\pi}}_{\overline{\mathbf{P}}} \rangle$-*Lipschitz* if for all $\bar{s}_1, \bar{s}_2 \in \overline{\mathcal{S}}$,

$$\left|\overline{\mathcal{R}}_{\bar{\pi}}(\bar{s}_1) - \overline{\mathcal{R}}_{\bar{\pi}}(\bar{s}_2)\right| \leqslant K^{\bar{\pi}}_{\overline{\mathcal{R}}} d_{\overline{\mathcal{S}}}(\bar{s}_1, \bar{s}_2),$$

$$W_{d_{\overline{\mathcal{S}}}}\left(\overline{\mathbf{P}}_{\bar{\pi}}(\cdot \mid \bar{s}_1), \overline{\mathbf{P}}_{\bar{\pi}}(\cdot \mid \bar{s}_2)\right) \leqslant K^{\bar{\pi}}_{\overline{\mathbf{P}}} d_{\overline{\mathcal{S}}}(\bar{s}_1, \bar{s}_2).$$

Local losses. Let $\xi \in \mathcal{P}(\mathcal{S} \times \mathcal{A})$, *local losses* are defined as:

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

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

where (i) $\mathcal{R}(s, \bar{a})$, (ii) $\mathbf{P}(\cdot \mid s, \bar{a})$, and (iii) $\phi\mathbf{P}(\cdot \mid s, \bar{a})$ are shorthand for (i) $\mathcal{R}(s, \psi(s, \bar{a}))$, (ii) $\mathbf{P}(\cdot \mid s, \psi(s, \bar{a}))$, and (iii) the distribution over $\overline{\mathcal{S}}$ of sampling $s' \sim \mathbf{P}(\cdot \mid s, \bar{a})$ and then embedding $\bar{s}' = \phi(s)$.

Assuming $\overline{\mathcal{M}}$ is discrete, all $W_{d_{\overline{\mathcal{S}}}}$ terms can be replaced by d_{TV} since Wasserstein coincides with TV when using the discrete metric. In that case, (optimal) constants $K^{\bar{\pi}}_{\overline{\mathcal{R}}}$ and $K^{\bar{\pi}}_{\overline{\mathbf{P}}}$ can be computed in polynomial time in $\overline{\mathcal{M}}$ for any $\bar{\pi} \in \overline{\Pi}^{\mathrm{ml}}$.

Henceforth, we make the following assumptions.

Assumption 3.1. *MDP $\mathcal{M}$ is ergodic.*

Assumption 3.2. *Rewards of $\mathcal{M}$ are scaled in the interval $[-\frac{1}{2}, \frac{1}{2}]$, i.e., $\mathcal{R} \colon \mathcal{S} \times \mathcal{A} \to [-\frac{1}{2}, \frac{1}{2}]$.*

Assumption 3.3. *The embedding function preserves the labels, i.e., $\phi(s) = \bar{s} \implies \ell(s) = \bar{\ell}(\bar{s})$ for $s \in \mathcal{S}$, $\bar{s} \in \overline{\mathcal{S}}$.*

Seemingly restrictive at first glance, Assumption 3.1 is compliant with RL environments and a wide range of continuous learning tasks (every episodic RL process is ergodic, see Huang 2020). Discarding it restricts the upcoming guarantees to BSCCs. Assumption 3.2 basically requires rewards to be bounded and re-scalable in this interval. This is a reasonable assumption in practice and re-scaling is straightforward if bounds are known (otherwise, dynamical re-scaling is still feasible but complicates the implementation). In Sect. 4.2, we show that Assumption 3.3 can be made trivial. Note that our approach requires Assumption 3.2 and 3.3.

3.1 Bisimulation and Value Difference Bounds

We aim now at formally checking whether the latent space model offers a good abstraction of the original MDP $\mathcal{M}$. To do so, we present bounds that link the two MDPs. We extend the bounds from Gelada et al. (2019) to discrete spaces while additionally taking into account state labels and discounted reachability events. Moreover, we present new *bisimulation* bounds in the local setting.

Bisimulation. A *(probabilistic) bisimulation* is a behavioral equivalence between states. Formally, a bisimulation on $\mathcal{M}$ is an equivalence relation B_{Φ} such that for all $s_1, s_2 \in \mathcal{S}$ and $\Phi \subseteq \{\mathcal{R}, \ell\}$, $s_1 B_{\Phi} s_2$ iff $\mathbf{P}(T \mid s_1, a) = \mathbf{P}(T \mid s_2, a)$, $\ell(s_1) = \ell(s_2)$ if $\ell \in \Phi$, and $\mathcal{R}(s_1, a) = \mathcal{R}(s_2, a)$ if $\mathcal{R} \in \Phi$, for each action $a \in \mathcal{A}$, and (Borel measurable) equivalence class $T \in \mathcal{S}/B_{\Phi}$. Properties of bisimulation include trace, trajectory, and value equivalence (Larsen and Skou 1989; Givan, Dean, and Greig 2003). The relation can be extended to compare two MDPs (in our case $\mathcal{M}$ and $\overline{\mathcal{M}}$) by considering the disjoint union of their state space. We denote the largest bisimulation relation by $\sim_{\Phi}$.

Pseudometrics. Desharnais et al. (2004) introduced *bisimulation pseudometrics* for continuous Markov processes, generalizing the notion of bisimilariy by assigning a *bisimilarity distance* between states. A *pseudometric* $\tilde{d}$ satisfies symmetry and the triangle inequality.

Probabilistic bisimilarity can be characterized by a logical family of functional expressions derived from a logic $\mathcal{L}$. More specifically, given a policy $\pi \in \Pi^{\mathrm{ml}}_{\mathcal{M}}$, we consider a family $\mathcal{F}^{\mathcal{L}}_{\gamma}(\pi)$ of real-valued functions f, parameterized by the discount factor γ and defining the semantics of $\mathcal{L}$ in $\mathcal{M}_{\pi}$. The related pseudometric $\tilde{d}_{\pi}$ is defined as: $\tilde{d}_{\pi}(s_1, s_2) = \sup_{f \in \mathcal{F}^{\mathcal{L}}_{\gamma}(\pi)} |f(s_1) - f(s_2)|$, for all $s_1, s_2 \in \mathcal{S}$. We distinguish between pseudometrics $\tilde{d}^{\mathcal{R}}_{\pi}$, characterized by functional expressions including rewards, and $\tilde{d}^{\ell}_{\pi}$, whose functional expressions are based on state labels. Let $\tilde{P}$ be the space of pseudometrics on $\mathcal{S}$ and $\varphi \in \{\mathcal{R}, \ell\}$. Define $\Delta \colon \tilde{P} \to \tilde{P}$ so that $\Delta(d^{\varphi}_{\pi})(s_1, s_2)$ is $(1-\gamma)\left|\mathcal{R}_{\pi}(s_1) - \mathcal{R}_{\pi}(s_2)\right| \cdot \mathbf{1}_{\{\mathcal{R}\}}(\varphi) + M$ where:

$$M = \max \begin{cases} \gamma W_{d^{\varphi}_{\pi}}\left(\mathbf{P}_{\pi}(\cdot \mid s_1), \mathbf{P}_{\pi}(\cdot \mid s_2)\right), \\ \mathbf{1}_{\neq}(\ell(s_1), \ell(s_2)) \cdot \mathbf{1}_{\{\ell\}}(\varphi), \end{cases}$$

then $\tilde{d}^{\varphi}_{\pi}$ is its unique fixed point whose kernel is $\sim_{\{\varphi\}}$, i.e., $\tilde{d}^{\varphi}_{\pi}(s_1, s_2) = 0$ iff $s_1 \sim_{\{\varphi\}} s_2$ (van Breugel and Worrell 2001; Ferns, Precup, and Knight 2014), and $|V_{\pi}(s_1) - V_{\pi}(s_2)| \leqslant \tilde{d}^{\mathcal{R}}_{\pi}(s_1, s_2)/1-\gamma$ (Ferns, Panangaden, and Precup 2005).

Bisimulation distance bounds. We claim that the expected bisimulation distance between states and their latent abstraction is bounded by local losses. Fix $\bar{\pi} \in \overline{\Pi}^{\mathrm{ml}}$ and assume $\overline{\mathcal{M}}$ is discrete and $\langle K^{\bar{\pi}}_{\overline{\mathcal{R}}}, K^{\bar{\pi}}_{\overline{\mathbf{P}}} \rangle$-Lipschitz. Given the induced stationary distribution $\xi_{\bar{\pi}}$ in $\mathcal{M}$,

$$\mathop{\mathbb{E}}_{s \sim \xi_{\bar{\pi}}} \tilde{d}^{\mathcal{R}}_{\bar{\pi}}(s, \phi(s)) \leqslant L^{\xi_{\bar{\pi}}}_{\mathcal{R}} + \gamma L^{\xi_{\bar{\pi}}}_{\mathbf{P}} \frac{K^{\bar{\pi}}_{\overline{\mathcal{R}}}}{1 - \gamma K^{\bar{\pi}}_{\overline{\mathbf{P}}}},$$

$$\mathop{\mathbb{E}}_{s \sim \xi_{\bar{\pi}}} \tilde{d}^{\ell}_{\bar{\pi}}(s, \phi(s)) \leqslant \frac{\gamma L^{\xi_{\bar{\pi}}}_{\mathbf{P}}}{1 - \gamma}. \tag{1}$$

The result provides us a general way to asses the quality of our latent abstraction: the bisimulation distance between states and their embedding is guaranteed to be small in average whenever local losses are small. Moreover, this allows us to bound the bisimulation distance between states with same representation by local losses: for any states $s_1, s_2 \in \mathcal{S}$ with $\phi(s_1) = \phi(s_2)$,

$$\tilde{d}^{\mathcal{R}}_{\bar{\pi}}(s_1, s_2) \leqslant \left[L^{\xi_{\bar{\pi}}}_{\mathcal{R}} + \frac{\gamma L^{\xi_{\bar{\pi}}}_{\mathbf{P}} K^{\bar{\pi}}_{\overline{\mathcal{R}}}}{1 - \gamma K^{\bar{\pi}}_{\overline{\mathbf{P}}}} \right] \big(\xi^{-1}_{\bar{\pi}}(s_1) + \xi^{-1}_{\bar{\pi}}(s_2) \big),$$

$$\tilde{d}^{\ell}_{\bar{\pi}}(s_1, s_2) \leqslant \frac{\gamma L^{\xi_{\bar{\pi}}}_{\mathbf{P}}}{1 - \gamma} \big(\xi^{-1}_{\bar{\pi}}(s_1) + \xi^{-1}_{\bar{\pi}}(s_2) \big). \tag{2}$$

Value difference bounds. Considering discounted returns or a specific event, the quality of the latent abstraction induced by ϕ and ψ can be in particular formalized by means of *value difference bounds*. These bounds can be intuitively derived by taking the value function as a real-valued function from $\mathcal{F}^{\mathcal{L}}_{\gamma}(\pi)$. Let $\mathsf{C}, \mathsf{T} \subseteq \mathbf{AP}$, $\varphi \in \{ \mathsf{C} \mathcal{U} \mathsf{T}, \Diamond \mathsf{T} \}$ and $K_{\overline{V}} = \min\big({}^{|\overline{\mathcal{R}}^{\star}_{\bar{\pi}}|}/_{1-\gamma}, {}^{K^{\bar{\pi}}_{\overline{\mathcal{R}}}}/_{1-\gamma K^{\bar{\pi}}_{\overline{\mathbf{P}}}} \big)$, then,

$$\mathop{\mathbb{E}}_{s,\bar{a} \sim \xi_{\bar{\pi}}} \left| Q_{\bar{\pi}}(s, \bar{a}) - \overline{Q}_{\bar{\pi}}(\phi(s), \bar{a}) \right| \leqslant \frac{L^{\xi_{\bar{\pi}}}_{\mathcal{R}} + \gamma K_{\overline{V}} L^{\xi_{\bar{\pi}}}_{\mathbf{P}}}{1 - \gamma},$$

$$\mathop{\mathbb{E}}_{s,\bar{a} \sim \xi_{\bar{\pi}}} \left| Q^{\varphi}_{\bar{\pi}}(s, \bar{a}) - \overline{Q}^{\varphi}_{\bar{\pi}}(\phi(s), \bar{a}) \right| \leqslant \frac{\gamma L^{\xi_{\bar{\pi}}}_{\mathbf{P}}}{1 - \gamma}. \tag{3}$$

Moreover, for any states $s_1, s_2 \in \mathcal{S}$ with $\phi(s_1) = \phi(s_2)$,

$$|V_{\bar{\pi}}(s_1) - V_{\bar{\pi}}(s_2)| \leqslant \frac{L^{\xi_{\bar{\pi}}}_{\mathcal{R}} + \gamma K_{\overline{V}} L^{\xi_{\xi_{\bar{\pi}}}}_{\mathbf{P}}}{1 - \gamma} \big(\xi^{-1}_{\bar{\pi}}(s_1) + \xi^{-1}_{\bar{\pi}}(s_2) \big),$$

$$|V^{\varphi}_{\bar{\pi}}(s_1) - V^{\varphi}_{\bar{\pi}}(s_2)| \leqslant \frac{\gamma L^{\xi_{\bar{\pi}}}_{\mathbf{P}}}{1 - \gamma} \big(\xi^{-1}_{\bar{\pi}}(s_1) + \xi^{-1}_{\bar{\pi}}(s_2) \big). \tag{4}$$

Intuitively, when local losses are sufficiently small, then (i) the expected value difference of states and their embeddings that are likely to be seen under a latent policy is also small, and (ii) states with the same embedding have close values.

3.2 Checking the Quality of the Abstraction

While bounding the difference between values offered by $\bar{\pi} \in \overline{\Pi}^{\mathrm{ml}}$ in $\mathcal{M}$ and $\overline{\mathcal{M}}$ is theoretically possible using $L^{\xi_{\bar{\pi}}}_{\mathcal{R}}$ and $L^{\xi_{\bar{\pi}}}_{\mathbf{P}}$, we need to accurately approximate these losses from samples to further offer practical guarantees (recall that $\mathcal{M}$ is unknown). Although the agent is able to produce execution traces by interacting with $\mathcal{M}$, estimating the expectation over the Wasserstein is intractable. Intuitively, even if approximating Wasserstein from samples is possible (e.g., Genevay et al. 2019), this would require access to a generative model for $\mathbf{P}(\cdot \mid s, a)$ (e.g., Kearns, Mansour, and Ng 2002) from which we would have to draw a sufficient number of samples for each s, a drawn from $\xi_{\bar{\pi}}$ to then be able to estimate the expectation. Gelada et al. (2019) overcome this issue by assuming a deterministic MDP, which allows optimizing an approximation of $L^{\xi_{\bar{\pi}}}_{\mathbf{P}}$ through gradient descent. To deal with general MDPs, we study an upper bound on $L^{\xi_{\bar{\pi}}}_{\mathbf{P}}$ that can be efficiently approximated from samples:

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

where $\xi_{\bar{\pi}}(s, \bar{a}, s') = \xi_{\bar{\pi}}(s, \bar{a}) \cdot \mathbf{P}(s' \mid s, \bar{a})$ and $\phi(\bar{s} \mid s) = \mathbf{1}_{=}(\phi(s), \bar{s})$. We now provide *probably approximately correct* (PAC) guarantees for estimating local losses for discrete latent models, derived from the Hoeffding's inequalities.

Lemma 3.4. *Suppose $\overline{\mathcal{M}}$ is discrete and the agent interacts with $\mathcal{M}$ by executing $\bar{\pi} \in \overline{\Pi}$, thus producing $\langle s_{0:T}, \bar{a}_{0:T-1}, r_{0:T-1} \rangle \sim \xi_{\bar{\pi}}$. Let $\varepsilon, \delta \in]0, 1[$ and denote by*

$$\hat{L}^{\xi_{\bar{\pi}}}_{\mathcal{R}} = \frac{1}{T} \sum_{t=0}^{T-1} \left| r_t - \overline{\mathcal{R}}(\phi(s_t), \bar{a}_t) \right|,$$

$$\hat{L}^{\xi_{\bar{\pi}}}_{\mathbf{P}} = \frac{1}{T} \sum_{t=0}^{T-1} \left[1 - \overline{\mathbf{P}}(\phi(s_{t+1}) \mid \phi(s_t), \bar{a}_t) \right].$$

Then after $T \geqslant \left\lceil -\log\left(\frac{\delta}{4}\right) / 2\varepsilon^2 \right\rceil$ steps, $\left| L^{\xi_{\bar{\pi}}}_{\mathcal{R}} - \hat{L}^{\xi_{\bar{\pi}}}_{\mathcal{R}} \right| \leqslant \varepsilon$, $\left| \dot{L}^{\xi_{\bar{\pi}}}_{\mathbf{P}} - \hat{L}^{\xi_{\bar{\pi}}}_{\mathbf{P}} \right| \leqslant \varepsilon$ with probability at least $1 - \delta$.

This yields the following Theorem, finally allowing to check the abstraction quality as a bounded value difference.

Theorem 3.5. *Let $\bar{\pi} \in \overline{\Pi}^{\mathrm{ml}}$ and assume $\overline{\mathcal{M}}$ is discrete and $\langle K^{\bar{\pi}}_{\overline{\mathcal{R}}}, K^{\bar{\pi}}_{\overline{\mathbf{P}}} \rangle$-Lipschitz. Let $\mathsf{C}, \mathsf{T} \subseteq \mathbf{AP}$, $\varphi \in \{ \mathsf{C} \mathcal{U} \mathsf{T}, \Diamond \mathsf{T} \}$, and $K_{\overline{V}} = \min\big({}^{|\overline{\mathcal{R}}^{\star}_{\bar{\pi}}|}/_{1-\gamma}, {}^{K^{\bar{\pi}}_{\overline{\mathcal{R}}}}/_{1-\gamma K^{\bar{\pi}}_{\overline{\mathbf{P}}}} \big)$. Let $\varepsilon, \delta \in]0, 1[$ and $\xi_{\bar{\pi}}$ be the stationary distribution of $\mathcal{M}_{\bar{\pi}}$. Then, after $T \geqslant \left\lceil -\log\left(\frac{\delta}{4}\right) \left(1 + \gamma K_{\overline{V}}\right)^2 / 2\varepsilon^2 (1-\gamma)^2 \right\rceil$ interaction steps through $\xi_{\bar{\pi}}$,*

$$\mathop{\mathbb{E}}_{s,\bar{a} \sim \xi_{\bar{\pi}}} \left| Q_{\bar{\pi}}(s, \bar{a}) - \overline{Q}_{\bar{\pi}}(\phi(s), \bar{a}) \right| \leqslant \frac{\hat{L}^{\xi_{\bar{\pi}}}_{\mathcal{R}} + \gamma K_{\overline{V}} \hat{L}^{\xi_{\bar{\pi}}}_{\mathbf{P}}}{1 - \gamma} + \varepsilon,$$

$$\mathop{\mathbb{E}}_{s,\bar{a} \sim \xi_{\bar{\pi}}} \left| Q^{\varphi}_{\bar{\pi}}(s, \bar{a}) - \overline{Q}^{\varphi}_{\bar{\pi}}(\phi(s), \bar{a}) \right| \leqslant \frac{\gamma \hat{L}^{\xi_{\bar{\pi}}}_{\mathbf{P}}}{1 - \gamma} + \frac{\gamma \varepsilon}{1 + \gamma K_{\overline{V}}}$$

with probability at least $1 - \delta$.

4 Variational Markov Decision Processes

We now provide a framework based on *variational autoencoders* (Kingma and Welling 2014) that allows us to learn a discrete latent space model of $\mathcal{M}$ through the interaction of the agent executing a pre-learned RL policy $\pi \in \Pi^{\mathrm{ml}}_{\mathcal{M}}$ with the environment. Concretely, we seek a discrete latent space model $\langle \overline{\mathcal{M}}_{\theta}, \phi_{\iota}, \psi_{\iota,\theta} \rangle$ such that $\overline{\mathcal{M}}_{\theta} = \langle \overline{\mathcal{S}}, \overline{\mathcal{A}}, \overline{\mathbf{P}}_{\theta}, \overline{\mathcal{R}}_{\theta}, \bar{\ell}_{\theta}, \mathbf{AP}, \bar{s}_I \rangle$. We propose to learn the parameters $\langle \iota, \theta \rangle$ of an *encoder* Q_{ι} and a *behavioral model* P_{θ} from which we can retrieve (i) the embedding functions ϕ_{ι} and $\psi_{\iota,\theta}$, (ii) the latent MDP components $\overline{\mathbf{P}}_{\theta}$, $\overline{\mathcal{R}}_{\theta}$, and $\bar{\ell}_{\theta}$, and (iii) a latent policy $\bar{\pi}_{\theta} \in \overline{\Pi}^{\mathrm{ml}}$, via $\min_{\theta} D(\mathcal{M}_{\pi}, P_{\theta})$, where D is a discrepancy measure. Intuitively, the end goal is to learn (i) a discrete representation of $\mathcal{S}$ and $\mathcal{A}$ (ii) to mimic the behaviors of the original MDP over the induced latent spaces, thus yielding a latent MDP with a bisimulation distance close to $\mathcal{M}$, and (iii) to *distill* π into $\bar{\pi}_{\theta}$. In the following, we distinguish the case where we only learn $\overline{\mathcal{S}}$, with $\overline{\mathcal{A}} = \mathcal{A}$ (in that case, $\mathcal{A}$ is assumed to be discrete), and the one where we additionally need to discretize the set of actions and learn $\overline{\mathcal{A}}$.

4.1 Evidence Lower Bound

In this work, we focus on the case where D_{KL} is used as discrepancy measure: the goal is optimizing $\min_\theta D_{\mathrm{KL}}(\mathcal{M}_\pi \parallel P_\theta)$ or equivalently maximizing the marginal *log-likelihood of traces* of $\mathcal{M}$, i.e., $\mathbb{E}_{\hat{\tau}\sim\mathcal{M}_\pi}[\log P_\theta(\hat{\tau})]$, where

$$P_\theta(\hat{\tau}) = \int_{Traj_{\overline{\mathcal{M}}_\theta}} P_\theta(\hat{\tau} \mid z_{0:T})\, d\overline{\mathbf{P}}_{\overline{\pi}_\theta}(z_{0:T}), \tag{5}$$

$\hat{\tau} = \langle s_{0:T}, a_{0:T-1}, r_{0:T-1}, l_{0:T}\rangle$, $z \in \mathcal{Z}$ with $\mathcal{Z} = \overline{\mathcal{S}}$ if $\overline{\mathcal{A}} = \mathcal{A}$ and $\mathcal{Z} = \overline{\mathcal{S}} \times \overline{\mathcal{A}}$ otherwise, $\overline{\mathbf{P}}_{\overline{\pi}_\theta}(\bar{s}_{0:T}) = \prod_{t=0}^{T-1} \overline{\mathbf{P}}_{\overline{\pi}_\theta}(\bar{s}_{t+1} \mid \bar{s}_t)$, and $\overline{\mathbf{P}}_{\overline{\pi}_\theta}(\bar{s}_{0:T}, \bar{a}_{0:T-1}) = \prod_{t=0}^{T-1} \overline{\pi}_\theta(\bar{a}_t \mid \bar{s}_t) \cdot \overline{\mathbf{P}}_\theta(\bar{s}_{t+1} \mid \bar{s}_t, \bar{a}_t)$. The dependency of $\hat{\tau}$ on $\overline{\mathcal{Z}}$ in Eq. 5 is made explicit by the law of total probability.

Optimizing $\mathbb{E}_{\hat{\tau}\sim\mathcal{M}_\pi}[\log P_\theta(\hat{\tau})]$ through Eq. 5 is typically intractable (Kingma and Welling 2014). To overcome this, we use an encoder $Q_\iota(z_{0:T} \mid \hat{\tau})$ to set up a *lower bound* on the log-likelihood of produced traces, often referred to as *evidence lower bound* (ELBO, Hoffman et al. 2013):

$$\begin{aligned}&\log P_\theta(\hat{\tau}) - D_{\mathrm{KL}}(Q_\iota(\cdot \mid \hat{\tau}) \parallel P_\theta(\cdot \mid \hat{\tau}))\\ =&\mathop{\mathbb{E}}_{z_{0:T}\sim Q_\iota(\cdot\mid\hat{\tau})}[\log P_\theta(\hat{\tau} \mid z_{0:T})] - D_{\mathrm{KL}}\big(Q_\iota(\cdot \mid \hat{\tau}) \parallel \overline{\mathbf{P}}_{\overline{\pi}_\theta}\big)\end{aligned}$$

The purpose of optimizing the ELBO is twofold. First, this allows us learning ϕ_ι via (i) $Q_\iota(\bar{s}_{0:T} \mid \hat{\tau}) = \prod_{t=0}^{T} \phi_\iota(\bar{s}_t \mid s_t)$ or (ii) $Q_\iota(\bar{s}_{0:T}, \bar{a}_{0:T-1} \mid \hat{\tau}) = Q_\iota(\bar{a}_{0:T-1} \mid \bar{s}_{0:T}, \hat{\tau}) \cdot Q_\iota(\bar{s}_{0:T} \mid \hat{\tau})$, where $Q_\iota(\bar{a}_{0:T-1} \mid \bar{s}_{0:T}, \hat{\tau}) = \prod_{t=0}^{T-1} Q_\iota^{\mathcal{A}}(\bar{a}_t \mid \bar{s}_t, a_t)$, $Q_\iota^{\mathcal{A}}$ being an action encoder. We assume here that encoding states and actions to latent spaces is independent of rewards. We additionally make them independent of $l_{0:T}$ by assuming that ℓ is known. This allows ϕ_ι to encode states and their labels directly into the latent space (cf. Sect. 4.2).

Second, we assume the existence of latent reward and label models, i.e., $P_\theta^{\mathcal{R}}$ and P_θ^{ℓ}, allowing to recover respectively $\overline{\mathcal{R}}_\theta$ and $\bar{\ell}_\theta$, as well as a *generative model* $P_\theta^{\mathcal{G}}$, enabling the reconstruction of states and actions. This allows decomposing the behavioral model P_θ into:

$$\begin{aligned}&P_\theta(s_{0:T}, a_{0:T-1}, r_{0:T-1}, l_{0:T} \mid z_{0:T})\\ =&P_\theta^{\mathcal{G}}(s_{0:T} \mid \bar{s}_{0:T}) \cdot P_\theta^{\mathcal{G}}(a_{0:T-1} \mid z_{0:T})\\ &\cdot P_\theta^{\mathcal{R}}(r_{0:T-1} \mid \bar{s}_{0:T}, \bar{a}_{0:T-1}) \cdot P_\theta^{\ell}(l_{0:T} \mid \bar{s}_{0:T}),\end{aligned}$$

$$\text{where } P_\theta^{\mathcal{G}}(s_{0:T} \mid \bar{s}_{0:T}) = \prod_{t=0}^{T} P_\theta^{\mathcal{G}}(s_t \mid \bar{s}_t),$$

$$P_\theta^{\mathcal{G}}(a_{0:T-1} \mid z_{0:T-1}) = \begin{cases}\prod_{t=0}^{T-1} \overline{\pi}_\theta(a_t \mid \bar{s}_t) \text{ if } \overline{\mathcal{A}} = \mathcal{A}\\ \prod_{t=0}^{T-1} \psi_\theta(a_t \mid \bar{s}_t, \bar{a}_t) \text{ else,}\end{cases}$$

$$P_\theta^{\mathcal{R}}(r_{0:T-1} \mid \bar{s}_{0:T}, \bar{a}_{0:T-1}) = \prod_{t=0}^{T-1} P_\theta^{\mathcal{R}}(r_t \mid \bar{s}_t, \bar{a}_t), \text{and}$$

$$P_\theta^{\ell}(l_{0:T} \mid \bar{s}_{0:T}) = \prod_{t=0}^{T} P_\theta^{\ell}(l_t \mid \bar{s}_t).$$

Model ψ_θ allows learning the action embedding function via $\psi_{\iota,\theta}(a \mid s, \bar{a}) = \mathbb{E}_{\bar{s}\sim\phi_\iota(\cdot\mid s)}\, \psi_\theta(a \mid \bar{s}, \bar{a})$ for all $s \in \mathcal{S}, a \in \mathcal{A}, \bar{a} \in \overline{\mathcal{A}}$. We also argue that a *perfect reconstruction* of labels is possible, i.e., $P_\theta^{\ell}(l_{0:T} \mid \bar{s}_{0:T}) = 1$, due to the labels being encoded into the latent space. From now on, we thus omit the label term.

Deterministic embedding functions ϕ_ι, $\psi_{\iota,\theta}$ and $\overline{\mathcal{R}}_\theta$ can finally be obtained by taking the mode of their distribution.

Back to the local setting. Taking Assumption 3.1 into account, drawing multiple finite traces $\hat{\tau} \sim \mathcal{M}_\pi$ can be seen as a continuous interaction with $\mathcal{M}$ along an infinite trace (Huang 2020). This observation allows us to formulate the ELBO in the local setting and connect to local losses:

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

where $\mathbf{D}$ and $\mathbf{R}$ denote respectively the *distortion* and *rate* of the variational model (Alemi et al. 2018), given by

$$\mathbf{D}_{\iota,\theta} = -\begin{cases}\mathop{\mathbb{E}}_{\substack{s,a,r,s'\sim\xi_\pi\\ \bar{s},\bar{s}'\sim\phi_\iota(\cdot\mid s,s')}} \big[\log P_\theta^{\mathcal{G}}(s' \mid \bar{s}') + \log \overline{\pi}_\theta(a \mid \bar{s}) + \\ \qquad \log P_\theta^{\mathcal{R}}(r \mid \bar{s}, a)\big] \text{ if } \overline{\mathcal{A}} = \mathcal{A},\\ \mathop{\mathbb{E}}_{\substack{s,a,r,s'\sim\xi_\pi\\ \bar{s},\bar{s}'\sim\phi_\iota(\cdot\mid s,s')\\ \bar{a}\sim Q_\iota^{\mathcal{A}}(\cdot\mid\bar{s},a)}} \big[\log P_\theta^{\mathcal{G}}\big(s' \mid \bar{s}'\big) + \\ \qquad \log \psi_\theta(a \mid \bar{s}, \bar{a}) + \\ \qquad \log P_\theta^{\mathcal{R}}(r \mid \bar{s}, \bar{a})\big] \text{ else, and}\end{cases}$$

$$\mathbf{R}_{\iota,\theta} = \begin{cases}\mathop{\mathbb{E}}_{\substack{s,a,s'\sim\xi_\pi\\ \bar{s}\sim\phi_\iota(\cdot\mid s)}} D_{\mathrm{KL}}\big(\phi_\iota\big(\cdot \mid s'\big) \parallel \overline{\mathbf{P}}_{\overline{\pi}_\theta}(\cdot \mid \bar{s})\big) \text{ if } \overline{\mathcal{A}} = \mathcal{A},\\ \mathop{\mathbb{E}}_{\substack{s,a,s'\sim\xi_\pi\\ \bar{s}\sim\phi_\iota(\cdot\mid s)\\ \bar{a}\sim Q_\iota^{\mathcal{A}}(\cdot\mid\bar{s},a)}} \big[D_{\mathrm{KL}}\big(\phi_\iota\big(\cdot \mid s'\big) \parallel \overline{\mathbf{P}}_\theta(\cdot \mid \bar{s}, \bar{a})\big) + \\ \qquad D_{\mathrm{KL}}\big(Q_\iota^{\mathcal{A}}(\cdot \mid \bar{s}, a) \parallel \overline{\pi}_\theta(\cdot \mid \bar{s})\big)\big] \text{ else.}\end{cases}$$

We omit the subscripts when the context is clear. The optimization of $\mathrm{ELBO}\big(\overline{\mathcal{M}}_\theta, \phi_\iota, \psi_{\iota,\theta}\big)$ allows for an indirect optimization of the local losses through their variational versions: (i) $L_{\mathcal{R}}^{\xi_{\overline{\pi}}}$ via the log-likelihood of rewards produced, and (ii) $\dot{L}_{\mathbf{P}}^{\xi_{\overline{\pi}}}$ where we change the Wasserstein term to the KL divergence. Note that this last change means we do not necessarily obtain the theoretical guarantees on the quality of the abstraction via its optimization. Nevertheless, our experiments indicate KL divergence is a good proxy of the Wasserstein term in practice. In particular, in the discrete setting, Wasserstein matches TV and one can relate the proxy with the original metric using the Pinkster's inequality.

4.2 VAE Distributions

Discrete distributions. We aim at learning *discrete* latent spaces $\overline{\mathcal{S}}$ and $\overline{\mathcal{A}}$, the distributions ϕ_ι, $Q_\iota^{\mathcal{A}}$, $\overline{\mathbf{P}}_\theta$, and $\overline{\pi}_\theta$ are thus supposed to be discrete. Two main challenges arise: (i) gradient descent is not applicable to learn ι and θ due to the discontinuity of $\overline{\mathcal{S}}$ and $\overline{\mathcal{A}}$, and (ii) sampling from these distributions must be a *derivable operation*. We overcome these by using *continuous relaxation of Bernoulli distributions* to learn a binary representation of the latent states, and the *Gumbel softmax trick* for the latent action space (Jang, Gu, and Poole 2017; Maddison, Mnih, and Teh 2017).

Labels. To enable $\log P_\theta^{\ell}(l_{0:T} \mid \bar{s}_{0:T}) = 0$, we linearly encode $\ell(s_t) = l_t$ into each $\bar{s}_t$ via ϕ_ι. Recall that labels are binary encoded, so we allocate them $|\mathbf{AP}|$ bits in $\overline{\mathcal{S}}$. Then, $\phi_\iota(\bar{s} \mid s) > 0$ implies $\ell(s) = \bar{\ell}_\theta(\bar{s})$, for all $s \in \mathcal{S}, \bar{s} \in \overline{\mathcal{S}}$, satisfying Assumption 3.3 if ϕ_ι is deterministic.

Decoders. For $P_\theta^{\mathcal{G}}$, ψ_θ, and $P_\theta^{\mathcal{R}}$, we learn the parameters of multivariate normal distributions. This further allows linking all $\bar{s} \in \overline{\mathcal{S}}$ to the parameters of $P_\theta^{\mathcal{G}}(\cdot \mid \bar{s})$ for explainability.

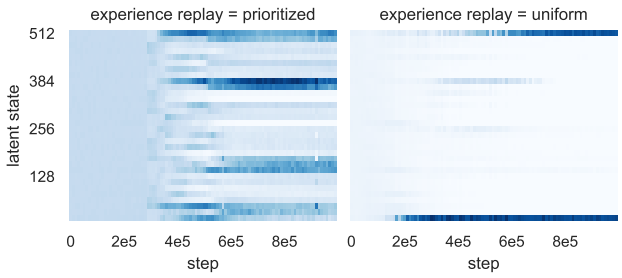


Figure 1: Latent space distribution along training steps for the CartPole environment. The intensity of the blue hue corresponds to the frequency of latent states produced by ϕ_ι during training. We compare a bucket-based prioritized against a simple uniform experience replay. The latent space learned via the uniform buffer collapses to two latent states.

4.3 Posterior Collapse

A common issue encountered while optimizing variational models via the ELBO is *posterior collapse*. Intuitively, this results in a degenerate local optimum where the model learns to ignore the latent space. With a discrete encoder, this translates into a deterministic mapping to a single latent state, regardless of the input. From an information-theoretic point of view, optimizing the ELBO gives way to a trade-off between the minimization of $\mathbf{R}$ and $\mathbf{D}$, where the feasible region is a convex set (Alemi et al. 2018). Posterior collapse occurs when $\mathbf{R} \approx 0$ (*auto-decoding limit*). On the other hand, one can achieve $\mathbf{D} \approx 0$ (*auto-encoding limit*) at the price of a higher rate.

Regularization terms. Various solutions have been proposed in the literature to prevent posterior collapse. They include *entropy regularization* (via $\alpha \in [0, \infty[$, Haarnoja et al. 2018; Dong et al. 2020) and *KL-scaling* (via $\beta \in [0, 1]$, e.g., Alemi et al. 2018), consisting in changing ELBO to $\langle \alpha, \beta \rangle$-ELBO $= -(\mathbf{D} + \beta \cdot \mathbf{R}) + \alpha \cdot H(Q_\iota)$, where $H(Q_\iota)$ denotes the entropy of an encoding distribution. We choose to measure the entropy of the *marginal* encoder, given by $Q_\iota(\bar{s}) = \mathbb{E}_{s \sim \xi_\pi} \phi_\iota(\bar{s} \mid s)$. Intuitively, this encourages the encoder to learn to make plenty use of the latent space. The parameter β allows to interpolate between auto-encoding and auto-decoding behavior, which is not possible with the standard ELBO objective.

A drawback of these methods is that we no longer optimize a lower bound on the log-likelihood of the input while optimizing $\langle \alpha, \beta \rangle$-ELBO. In practice, setting up annealing schemes for α and β allows to eventually recover ELBO and avoid posterior collapse ($\alpha = 0$ and $\beta = 1$ matches ELBO).

Prioritized replay buffers. To enable meaningful use of latent space to represent input state-actions pairs, Q_ι should learn to (i) exploit the entire latent space, and (ii) encode wisely states and actions of transitions yielding poor ELBO, being generally sensitive to bad representation embedding. This motivates us to use a *prioritized replay buffer* (Schaul et al. 2016) to store transitions and sample them when optimizing ELBO. Draw $\langle s, a, r, s' \rangle \sim \xi_\pi$ and let $p_{\langle s,a,r,s' \rangle}$ be its priority, we introduce the following priority functions.

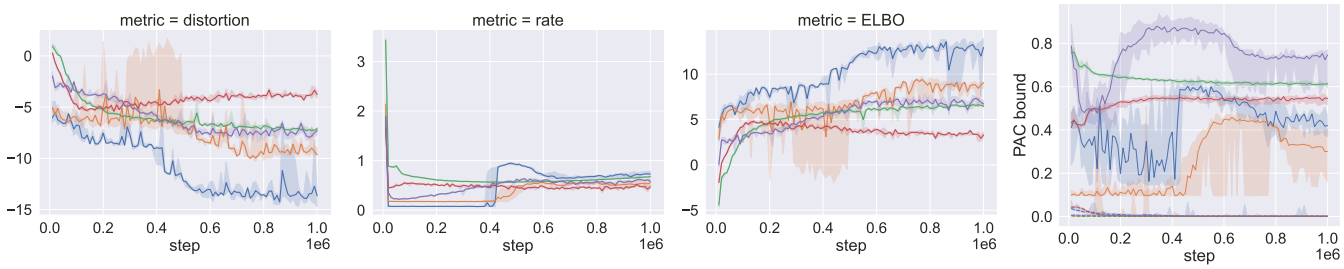
- *Bucket-based priority:* we partition the buffer in $|\bar{\mathcal{S}}|$ *buckets*. Let $N \in \mathbb{N}$ and $b \colon \bar{\mathcal{S}} \to \mathbb{N}$ be respectively step and latent state counters. At each step, let $\bar{s} \sim \phi_\iota(\cdot \mid s)$, we assign $p_{\langle s,a,r,s' \rangle}$ to $N / b(\bar{s})$, then increment N and $b(\bar{s})$ of one. This allows ϕ_ι to process states being infrequently visited under π and learn to fairly distribute $\bar{\mathcal{S}}$.
- *Loss-based priority:* we set $p_{\langle s,a,r,s' \rangle}$ to its individual transition loss, which enables to learn improving the representation of states and actions that yield poor ELBO.

5 Experiments

The goal of our experiments is to evaluate the quality of the latent space model learned and the policy distilled via our VAE-MDP framework. This evaluation consists of: an analysis of the training of the latent space model and the benefits of our method to avoid posterior collapse, assessing the quality of the abstraction learned via PAC local losses bounds, and testing the performance of the distilled policy. This allows to assess if the latent model learned yields a sound compression of the state-action space that retains the necessary information to optimize the return. We evaluate our method on classic OpenAI environments (Brockman et al. 2016) with (i) continuous states and discrete actions (CartPole, MountainCar, and Acrobot), and (ii) where both, states and actions, are continuous (Pendulum and LunarLander). We distill RL policies π learned via DQN (Mnih et al. 2015) for case (i), and SAC (Haarnoja et al. 2018) for case (ii).

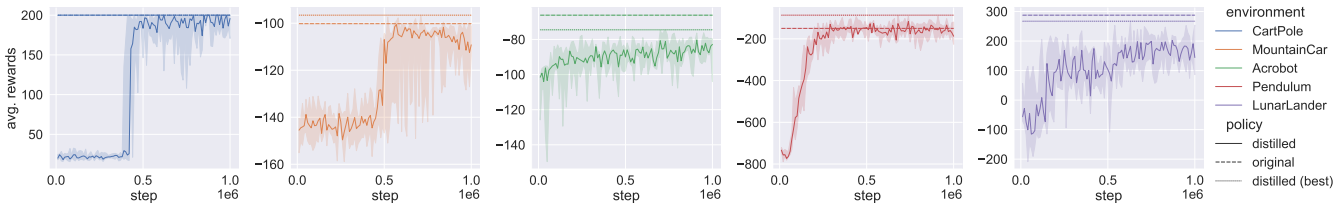
Latent spaces. Since latent spaces are trained to enable formal verification, we choose $\log_2 |\bar{\mathcal{S}}|$ and $|\bar{\mathcal{A}}|$ ranging from 9 bits and 2 actions (coarser) to 16 bits and 5 actions (finer) to make them tractable for model checkers — see Budde et al. (2020) for a performance comparison of modern tools for a range of instances with large model size. Depending on the property to verify, the latent space may have to be finer since (i) we need to reserve $|\mathbf{AP}|$ bits in the representation of $\bar{\mathcal{S}}$ for labels, and (ii) this allows the agent to take more precise decisions over a finer partition of the latent space. We then select the model with the best trade-off between abstraction quality (measured via $\hat{L}_{\mathcal{R}}^{\xi_{\bar{\pi}_\theta}}$ and $\hat{L}_{\mathbf{P}}^{\xi_{\bar{\pi}_\theta}}$) and performance (i.e., the return approximated by running $\bar{\pi}$ in $\mathcal{M}$).

ELBO optimization. In order to allow ELBO to be trained efficiently, posterior collapse has to be tackled from the very first stages of training. In fact, we found that the KL-scaling and entropy regularization annealing schemes (cf. Sect. 4.3) were necessary to avoid the latent space to collapse into a single state-action pair after only a few training steps. We found the most efficient to start with an autoencoder behavior ($\beta_0 = 0$) and a large entropy regularizer (e.g., $\alpha_0 = 10$) during the 10^4 first steps, and then anneal them via $\alpha_t = \alpha_0 \cdot (1-\tau)^t$, $\beta_t = 1-(1-\tau)^t$ (e.g., $\tau = 10^{-5}$) to fully recover the original ELBO in a second training phase. We also found that prioritized experience replays further prevent posterior collapse during training (cf. Fig. 1). Fig. 2a shows that training ELBO this way results in a stable learning procedure that successfully minimizes $\mathbf{D}$ while preventing an auto-decoding behavior by keeping $\mathbf{R}$ away from 0.



(a) $\mathbf{D}$, $\mathbf{R}$, and ELBO

(b) Local losses (PAC bound)



(c) Distilled policy evaluation

Figure 2: Plots reporting (2a) $\mathbf{D}$, $\mathbf{R}$, and ELBO (approximated over a large batch sampled from the replay buffer, using discrete latent distributions and averaged using importance sampling weights), (2b) PAC local losses approximation for $\varepsilon = 10^{-2}$, $\delta = 5 \cdot 10^{-3}$: solid lines stand for the transition loss and dashed lines for the reward loss, and (2c) the expected episode return (approximated by averaging over 30 episodes). For each environment, we train five different instances of our VAE with different random seeds, where the solid line corresponds to the median and the shaded interval to the interquartile range.

Local losses. We compute the PAC local losses bounds presented in Sect. 3 along training steps (Fig. 2b). When the policy is eventually distilled (Fig. 2c) and $\overline{\pi}_\theta$ achieves performance similar to those of π, the learning curves stabilize, while maximizing ELBO successfully allows minimizing the local losses. In every environment where we tested our method, a low reward loss is maintained while the transition loss reaches values in the interval $[1/5, 3/5]$ for most of the instances. These values are thus guaranteed to upper bound the expected bisimulation distance between $\mathcal{M}$ and $\overline{\mathcal{M}}$ as well as their value difference. We do not expect our approach to reach zero reward and transition loss though, since we pass from continuous to discrete spaces: the abstraction induced by our approach is always coarser than the original spaces, which translates in general to precision loss. This precision loss is often encoded through the discrete probability transitions. For instance, a state s which deterministically transitions to a close state s' in $\mathcal{M}$ such that $\phi_\iota(s) = \bar{s} = \phi_\iota(s')$ induces $\overline{\mathbf{P}}_{\overline{\pi}_\theta}(\bar{s} \mid \bar{s}) > 0$. Observe that the PAC computation of $\hat{L}_{\mathbf{P}}^{\xi_{\overline{\pi}_\theta}}$ is sensitive to the entropy of $\overline{\mathbf{P}}_\theta$, which thus may induce a residual transition loss. $L_{\mathbf{P}}^{\xi_{\overline{\pi}_\theta}}$ is on the contrary not sensitive to this. In practice, its value will thus often be lower than its approximated upper bound $\hat{L}_{\mathbf{P}}^{\xi_{\overline{\pi}_\theta}}$.

Policy distillation. The guarantees derived in Sect. 3 are only valid for $\overline{\pi}_\theta$, the policy under which formal properties can be verified. We further need evaluate if it achieves sound performance in the RL environment: checking the expected value difference via Thm. 3.5 is most significant when $V_{\overline{\pi}_\theta}(s)$ is worth in any state s likely to be reached in $\mathcal{M}_{\overline{\pi}_\theta}$. We compare the episode return achieved by $\overline{\pi}_\theta$ against π in the environment $\mathcal{M}$ (Fig. 2c). Our framework allows learning to improve $\overline{\pi}_\theta$ along training steps and eventually achieving the return of the original policy π. This illustrates that training our latent model enables distilling π into a latent policy $\overline{\pi}_\theta$ that achieves similar performance in $\mathcal{M}$.

6 Conclusion

In this work, we presented VAE-MDPs, a framework for learning discrete latent models of unknown, continuous-spaces environments with bisimulation guarantees. We detailed how such latent models can be learned by executing an RL policy in the environment and showed that the procedure yields a distilled version of the latter as a side effect. To provide the guarantees, we introduced new local losses bounds aimed at discrete latent spaces with their PAC-efficient approximation algorithm derived from the execution of the distilled policy. All this enables the verification of RL policies for unknown continuous MDPs.

Experimental results demonstrate the feasibility of our approach through the PAC bounds and the performance of the distilled policy achieved for various environments. Our tool can also be used to highlight the lack of robustness of input policies when the distillation fails.

Complementary to safe-RL approaches addressed via formal methods, we emphasize the ability of our tool to be coupled with such algorithms when no model of the environment is known a priori. The applicability of our method enables its use in future work for real-world case studies and more complex settings, such as multi-agent systems.

Acknowledgments

This research received funding from the Flemish Government (AI Research Program) and was supported by the DESCARTES iBOF project. G.A. Perez is also supported by the Belgian FWO "SAILor" project (G030020N).

References

Alemi, A. A.; Poole, B.; Fischer, I.; Dillon, J. V.; Saurous, R. A.; and Murphy, K. 2018. Fixing a Broken ELBO. In Dy, J. G.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, 159–168. PMLR.

Alshiekh, M.; Bloem, R.; Ehlers, R.; Könighofer, B.; Niekum, S.; and Topcu, U. 2018. Safe Reinforcement Learning via Shielding. In McIlraith, S. A.; and Weinberger, K. Q., eds., *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 2669–2678. AAAI Press.

Bacci, E.; and Parker, D. 2020. Probabilistic Guarantees for Safe Deep Reinforcement Learning. In Bertrand, N.; and Jansen, N., eds., *Formal Modeling and Analysis of Timed Systems - 18th International Conference, FORMATS 2020, Vienna, Austria, September 1-3, 2020, Proceedings*, volume 12288 of *LNCS*, 231–248. Springer.

Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. OpenAI Gym. *CoRR*, abs/1606.01540.

Budde, C. E.; Hartmanns, A.; Klauck, M.; Kretínský, J.; Parker, D.; Quatmann, T.; Turrini, A.; and Zhang, Z. 2020. On Correctness, Precision, and Performance in Quantitative Verification - QComp 2020 Competition Report. In Margaria, T.; and Steffen, B., eds., *Leveraging Applications of Formal Methods, Verification and Validation: Tools and Trends - 9th International Symposium on Leveraging Applications of Formal Methods, ISoLA 2020, Rhodes, Greece, October 20-30, 2020, Proceedings, Part IV*, volume 12479 of *LNCS*, 216–241. Springer.

Burden, J.; Siahroudi, S. K.; and Kudenko, D. 2021. Latent Property State Abstraction For Reinforcement learning. In *Proceedings of the AAMAS Workshop on Adaptive Learning Agents (ALA)*.

Carr, S.; Jansen, N.; and Topcu, U. 2020. Verifiable RNN-Based Policies for POMDPs Under Temporal Logic Constraints. In Bessiere, C., ed., *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, 4121–4127. ijcai.org.

Corneil, D. S.; Gerstner, W.; and Brea, J. 2018. Efficient ModelBased Deep Reinforcement Learning with Variational State Tabulation. In Dy, J. G.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, 1057–1066. PMLR.

Desharnais, J.; Gupta, V.; Jagadeesan, R.; and Panangaden, P. 2004. Metrics for labelled Markov processes. *Theor. Comput. Sci.*, 318(3): 323–354.

Dong, Z.; Seybold, B. A.; Murphy, K.; and Bui, H. H. 2020. Collapsed Amortized Variational Inference for Switching Nonlinear Dynamical Systems. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020*, volume 119 of *Proceedings of Machine Learning Research*, 2638–2647. PMLR.

Ferns, N.; Panangaden, P.; and Precup, D. 2005. Metrics for Markov Decision Processes with Infinite State Spaces. In *UAI '05, Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence, Edinburgh, Scotland, July 26-29, 2005*, 201–208. AUAI Press.

Ferns, N.; Precup, D.; and Knight, S. 2014. Bisimulation for Markov Decision Processes through Families of Functional Expressions. In van Breugel, F.; Kashefi, E.; Palamidessi, C.; and Rutten, J., eds., *Horizons of the Mind. A Tribute to Prakash Panangaden - Essays Dedicated to Prakash Panangaden on the Occasion of His 60th Birthday*, volume 8464 of *LNCS*, 319–342. Springer.

Freeman, C. D.; Ha, D.; and Metz, L. 2019. Learning to Predict Without Looking Ahead: World Models Without Forward Prediction. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, 5380–5391.

Gelada, C.; Kumar, S.; Buckman, J.; Nachum, O.; and Bellemare, M. G. 2019. DeepMDP: Learning Continuous Latent Space Models for Representation Learning. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, 2170–2179. PMLR.

Genevay, A.; Chizat, L.; Bach, F. R.; Cuturi, M.; and Peyré, G. 2019. Sample Complexity of Sinkhorn Divergences. In Chaudhuri, K.; and Sugiyama, M., eds., *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, 1574–1583. PMLR.

Givan, R.; Dean, T. L.; and Greig, M. 2003. Equivalence notions and model minimization in Markov decision processes. *Artif. Intell.*, 147(1-2): 163–223.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In Dy, J. G.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*,

volume 80 of *Proceedings of Machine Learning Research*, 1856–1865. PMLR.

Hartmanns, A.; and Hermanns, H. 2014. The Modest Toolset: An Integrated Environment for Quantitative Modelling and Verification. In Ábrahám, E.; and Havelund, K., eds., *Tools and Algorithms for the Construction and Analysis of Systems - 20th International Conference, TACAS 2014, Grenoble, France, April 5-13, 2014. Proceedings*, volume 8413 of *LNCS*, 593–598. Springer.

Hensel, C.; Junges, S.; Katoen, J.-P.; Quatmann, T.; and Volk, M. 2021. The probabilistic model checker Storm. *International Journal on Software Tools for Technology Transfer*.

Hoffman, M. D.; Blei, D. M.; Wang, C.; and Paisley, J. W. 2013. Stochastic variational inference. *J. Mach. Learn. Res.*, 14(1): 1303–1347.

Huang, B. 2020. Steady State Analysis of Episodic Reinforcement Learning. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Jang, E.; Gu, S.; and Poole, B. 2017. Categorical Reparameterization with Gumbel-Softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Jansen, N.; Könighofer, B.; Junges, S.; Serban, A.; and Bloem, R. 2020. Safe Reinforcement Learning Using Probabilistic Shields (Invited Paper). In Konnov, I.; and Kovács, L., eds., *31st International Conference on Concurrency Theory (CONCUR 2020)*, volume 171 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 3:1–3:16. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik. ISBN 978-3-95977-160-3.

Junges, S.; Jansen, N.; Dehnert, C.; Topcu, U.; and Katoen, J. 2016. Safety-Constrained Reinforcement Learning for MDPs. In Chechik, M.; and Raskin, J., eds., *Tools and Algorithms for the Construction and Analysis of Systems - 22nd International Conference, TACAS 2016, Eindhoven, The Netherlands, April 2-8, 2016, Proceedings*, volume 9636 of *LNCS*, 130–146. Springer.

Kearns, M. J.; Mansour, Y.; and Ng, A. Y. 2002. A Sparse Sampling Algorithm for Near-Optimal Planning in Large Markov Decision Processes. *Mach. Learn.*, 49(2-3): 193–208.

Kingma, D. P.; and Welling, M. 2014. Auto-Encoding Variational Bayes. In Bengio, Y.; and LeCun, Y., eds., *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Kwiatkowska, M.; Norman, G.; and Parker, D. 2011. PRISM 4.0: Verification of Probabilistic Real-time Systems. In Gopalakrishnan, G.; and Qadeer, S., eds., *Proc. 23rd International Conference on Computer Aided Verification (CAV'11)*, volume 6806 of *LNCS*, 585–591. Springer.

Larsen, K. G.; and Skou, A. 1989. Bisimulation Through Probabilistic Testing. In *Conference Record of the Sixteenth Annual ACM Symposium on Principles of Programming Languages, Austin, Texas, USA, January 11-13, 1989*, 344–352. ACM Press.

Lee, A. X.; Nagabandi, A.; Abbeel, P.; and Levine, S. 2020. Stochastic Latent Actor-Critic: Deep Reinforcement Learning with a Latent Variable Model. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Maddison, C. J.; Mnih, A.; and Teh, Y. W. 2017. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M. A.; Fidjeland, A.; Ostrovski, G.; Petersen, S.; Beattie, C.; Sadik, A.; Antonoglou, I.; King, H.; Kumaran, D.; Wierstra, D.; Legg, S.; and Hassabis, D. 2015. Human-level control through deep reinforcement learning. *Nat.*, 518(7540): 529–533.

Nowe, A. 1994. *Synthesis of "safe" fuzzy controllers based on reinforcement learning*. Ph.D. thesis, Vrije Universiteit Brussel.

Puterman, M. L. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. Wiley. ISBN 978-0-47161977-2.

Ren, T.; Niu, J.; Cui, J.; Ouyang, Z.; and Liu, X. 2021. An application of multi-objective reinforcement learning for efficient model-free control of canals deployed with IoT networks. *Journal of Network and Computer Applications*, 182: 103049.

Schaul, T.; Quan, J.; Antonoglou, I.; and Silver, D. 2016. Prioritized Experience Replay. In Bengio, Y.; and LeCun, Y., eds., *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Simão, T. D.; Jansen, N.; and Spaan, M. T. J. 2021. AlwaysSafe: Reinforcement Learning without Safety Constraint Violations during Training. In Dignum, F.; Lomuscio, A.; Endriss, U.; and Nowé, A., eds., *AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems, Virtual Event, United Kingdom, May 3-7, 2021*, 1226–1235. ACM.

Tsitsiklis, J. N. 1994. Asynchronous Stochastic Approximation and Q-Learning. *Mach. Learn.*, 16(3): 185–202.

Tsitsiklis, J. N.; and Roy, B. V. 1997. An analysis of temporal-difference learning with function approximation. *IEEE Trans. Autom. Control.*, 42(5): 674–690.

van Breugel, F.; and Worrell, J. 2001. Towards Quantitative Verification of Probabilistic Transition Systems. In Orejas, F.; Spirakis, P. G.; and van Leeuwen, J., eds., *Automata, Languages and Programming, 28th International Colloquium, ICALP 2001, Crete, Greece, July 8-12, 2001, Proceedings*, volume 2076 of *LNCS*, 421–432. Springer.