Exploiting Multiple Abstractions in Episodic RL via Reward Shaping

Roberto Cipollone[1], Giuseppe De Giacomo[1,2], Marco Favorito[3], Luca Iocchi[1], Fabio Patrizi[1]

[1]DIAG, Università degli Studi di Roma "La Sapienza", Italy
[2]Department of Computer Science, University of Oxford, U.K.
[3]Banca d'Italia, Italy
{cipollone, degiacomo, iocchi, patrizi}@diag.uniroma1.it
giuseppe.degiacomo@cs.ox.ac.uk, marco.favorito@bancaditalia.it

Abstract

One major limitation to the applicability of Reinforcement Learning (RL) to many practical domains is the large number of samples required to learn an optimal policy. To address this problem and improve learning efficiency, we consider a linear hierarchy of abstraction layers of the Markov Decision Process (MDP) underlying the target domain. Each layer is an MDP representing a coarser model of the one immediately below in the hierarchy. In this work, we propose a novel form of Reward Shaping where the solution obtained at the abstract level is used to offer rewards to the more concrete MDP, in such a way that the abstract solution guides the learning in the more complex domain. In contrast with other works in Hierarchical RL, our technique has few requirements in the design of the abstract models and it is also tolerant to modeling errors, thus making the proposed approach practical. We formally analyze the relationship between the abstract models and the exploration heuristic induced in the lower-level domain. Moreover, we prove that the method guarantees optimal convergence and we demonstrate its effectiveness experimentally.

1 Introduction

In Reinforcement Learning (RL), agents have no complete model available to predict the outcomes of their actions. Since coming up with a complete and faithful model of the world is generally difficult, this allows for the wide applicability of RL algorithms. Nonetheless, such a lack of knowledge also demands a significant number of interactions with the environment, before a (near-)optimal policy can be estimated. As a result, most of the successes of RL come from the digital world (e.g., video games, simulated environments), especially those in which a large number of samples can be easily generated, while applications to real environments, such as robotic scenarios, are still rare.

Many RL tasks are *goal-oriented*, which implies that when the goal states are *sparse*, so are the rewards. This is a well-known challenging scenario for RL, which further increases the amount of samples required. Unfortunately, sparse goal states are very common, as they may arise in simple tasks, such as reaching specific configurations in large state spaces, or, even in modest environments, for complex target behaviours, such as the successful completion of a sequence of smaller tasks (Brafman, De Giacomo, and Patrizi 2018; Icarte et al. 2018).

In order to improve sample efficiency, *Hierarchical* RL approaches (Hutsebaut-Buysse, Mets, and Latré 2022) have been proposed to decompose a complex problem into subtasks that are easier to solve individually. In this context, an *abstraction* is a simplified, coarser model that reflects the decomposition induced over the *ground*, more complex environment (Li, Walsh, and Littman 2006).

In this paper, we consider a linear hierarchy of abstractions of the Markov Decison Process (MDP) underlying the target domain. Each layer in this hierarchy is a simplified model, still represented as an MDP, of the one immediately below. A simple example is that of an agent moving in a map. The states of the ground MDP may capture the real pose of the agent, as continuous coordinates and orientation. The states of its abstraction, instead, may provide a coarser description, obtained by coordinate discretization, semantic map labelling (i.e., associating metric poses to semantic labels), or by projecting out state variables. Ultimately, such a compression corresponds to partitioning the ground state space into abstract states, implicitly defining a mapping from the former to the latter. The action spaces of the two models can also differ, in general, as they would include the actions that are best appropriate for each representation. Simulators are commonly used in RL and robotics. Often, simply though a different configuration of the same software, e.g. noiseless or ideal movement actions, it is possible to obtain a simplified environment which acts as an abstraction. Importantly, this simplified model also applies to a variety of tasks that may be defined over the same domain.

By taking advantage of the abstraction hierarchy, we devise an approach which allows any off-policy RL algorithm to efficiently explore the ground environment, while guaranteeing optimal convergence. The core intuition is that the value function $\bar{V}^*$ of the abstract MDP $\bar{\mathcal{M}}$ can be exploited to guide learning on the lower model $\mathcal{M}$. Technically, we adopt a variant of Reward Shaping (RS), whose potential is generated from $\bar{V}^*$. This way, when learning in $\mathcal{M}$, the agent is *biased* to visit first the states that correspond in $\bar{\mathcal{M}}$ to those that are preferred by the abstract policy, thus trying, in a sense, to replicate its behavior in $\mathcal{M}$. In order to guarantee effectiveness of the exploration bias, it is essential that the transitions of $\bar{\mathcal{M}}$ are good proxies for the dynamics

of $\mathcal{M}$. We characterize this relationship by identifying conditions under which the abstraction induces an exploration policy that is consistent with the ground domain.

The contributions of this work include: (i) the definition of a novel RS schema that allows for transferring the acquired experience from coarser models to the more concrete domains in the abstraction hierarchy; (ii) the derivation of a relationship between each abstraction and the exploration policy that is induced in the lower MDP; (iii) the identification of *abstract value approximation*, as a new condition to evaluate when an abstraction can be a good representative of ground MDP values; (iv) an experimental analysis showing that our approach significantly improves sample-efficiency and that modelling errors yield only a limited performance degradation.

2 Preliminaries

Notation Any total function $f : \mathcal{X} \to \mathcal{Y}$ induces a partition on its domain $\mathcal{X}$, such that two elements are in the same block iff $f(x) = f(x')$. We denote blocks of the partition by the elements of $\mathcal{Y}$, thus writing $x \in y$ instead of $x \in \{x' \mid x' \in \mathcal{X}, f(x') = y\}$. With $\Delta(\mathcal{Y})$, we denote the class of probability distributions over a set $\mathcal{Y}$. Also, $f : \mathcal{X} \to \Delta(\mathcal{Y})$ is a function returning a probability distribution (i.e., $f : \mathcal{X}, \mathcal{Y} \to [0, 1]$, with $\sum_{y \in \mathcal{Y}} f(x, y) = 1$, for all $x \in \mathcal{X}$).

Markov Decision Processes (MDPs) A Markov Decision Process (MDP) $\mathcal{M}$ is a tuple $\langle \mathcal{S}, \mathcal{A}, T, R, \gamma \rangle$, where $\mathcal{S}$ is a set of states, $\mathcal{A}$ is a set of actions, $T : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the probabilistic transition function, $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathcal{R}$ is the reward function (for $\mathcal{R} \coloneqq [r_-, r_+] \subset \mathbb{R}$), and $0 < \gamma < 1$ is the discount factor. A deterministic policy is a function $\rho : \mathcal{S} \to \mathcal{A}$. We follow the standard definitions for the value of a policy $V^\rho(s)$, as the expected sum of discounted returns, the value of an action $Q^\rho(s, a)$, and their respective optima $V^* = \max_{\rho \in P} V^\rho(s)$ and $Q^*(s, a)$ (Puterman 1994). We may also write $Q(s, \rho)$ to denote $V^\rho(s)$. Reinforcement Learning (RL) is the task of learning an optimal policy in an MDP with unknown T and R.

Definition 1. A RL learning algorithm is *off-policy* if, for any MDP $\mathcal{M}$ and experience $(\mathcal{ARS})^t$ generated by a stochastic exploration policy $\rho_e : \mathcal{S}, \mathbb{N} \to \Delta(\mathcal{A})$ such that $\forall s, t, a : \rho_e(s, t, a) > 0$, the algorithm converges to the optimal policy of $\mathcal{M}$, as $t \to \infty$.

Reward Shaping (RS) Reward Shaping (RS) is a technique for learning in MDPs with sparse rewards, which occur rarely during exploration. The purpose of RS is to guide the agent by exploiting some prior knowledge in the form of additional rewards: $R^s(s, a, s') \coloneqq R(s, a, s') + F(s, a, s')$, where F is the *shaping* function. In the classic approach, called *Potential-Based RS* (Ng, Harada, and Russell 1999) (simply called Reward Shaping from now on), the shaping function is defined in terms of a potential function, $\Phi : \mathcal{S} \to \mathbb{R}$, as:

$$F(s, a, s') \coloneqq \gamma\, \Phi(s') - \Phi(s) \tag{1}$$

If an infinite horizon is considered, this definition and its variants (Wiewiora, Cottrell, and Elkan 2003; Devlin and Kudenko 2012) guarantee that the set of optimal policies of $\mathcal{M}$ and $\mathcal{M}^s \coloneqq \langle \mathcal{S}, \mathcal{A}, T, R^s, \gamma \rangle$ coincide. Indeed, as shown by (Wiewiora 2003), the Q-learning algorithm over $\mathcal{M}^s$ performs the same updates as Q-learning over $\mathcal{M}$ with the modified Q-table initialization: $Q'_0(s, a) \coloneqq Q_0(s, a) + \Phi(s)$.

Options An option (Sutton, Precup, and Singh 1999), for an MDP $\mathcal{M}$, is a temporally-extended action, defined as $o = \langle \mathcal{I}_o, \rho_o, \beta_o \rangle$, where $\mathcal{I}_o \subseteq \mathcal{S}$ is an initiation set, $\rho_o : \mathcal{S} \to \mathcal{A}$ is the policy to execute, and $\beta_o : \mathcal{S} \to \{0, 1\}$ is a termination condition that, from the current state, computes whether the option should terminate. We write $Q^*(s, o)$ to denote the expected return of executing o until termination and following the optimal policy afterwards.

ϕ-Relative Options (Abel et al. 2020) Given an MDP $\mathcal{M}$ and a function $\phi : \mathcal{S} \to \bar{\mathcal{S}}$, an option $o = \langle \mathcal{I}_o, \rho_o, \beta_o \rangle$ of $\mathcal{M}$ is said to be *ϕ-relative* iff there is some $\bar{s} \in \bar{\mathcal{S}}$ such that:

$$\mathcal{I}_o = \{s \mid s \in \bar{s}\}, \quad \beta_o(s) = \mathbb{I}(s \notin \bar{s}), \quad \rho_o \in P_{\bar{s}} \tag{2}$$

where $\mathbb{I}(\xi)$ equals 1 if ξ is true, 0 otherwise, and $P_{\bar{s}}$ is the set of policies of the form $\rho_{\bar{s}} : \{s \mid s \in \bar{s}\} \to \mathcal{A}$. In other words, policies of ϕ-relative options are defined within some block $\bar{s}$ in the partition induced by ϕ and they must terminate exactly when the agent leaves the block where it was initiated.

3 Framework

Consider an environment in which experience is costly to obtain. This might be a complex simulation or an actual environment in which a physical robot is acting. This is our *ground* MDP $\mathcal{M}_0$ that we aim to solve while reducing the number of interactions with the environment. Instead of learning on this MDP directly, we choose to solve a simplified, related problem, that we call the *abstract* MDP. This idea is not limited to a single abstraction. Indeed, we consider a hierarchy of related MDPs $\mathcal{M}_0, \mathcal{M}_1, \ldots, \mathcal{M}_n$, of decreasing difficulty, where the experience acquired by an expert acting in $\mathcal{M}_i$ can be exploited to speed up learning in the previous one, $\mathcal{M}_{i-1}$.

Associated to each MDP abstraction $\mathcal{M}_i$, we also assume the existence of a *mapping* function $\phi_i : \mathcal{S}_i \to \mathcal{S}_{i+1}$, which projects states of $\mathcal{M}_i$ to states of its direct abstraction $\mathcal{M}_{i+1}$. This induces a partition over $\mathcal{S}_i$, where each block contains all states that are mapped through ϕ_i to a single state in $\mathcal{M}_{i+1}$. The existence of state mappings are a classic assumption in Hierarchical RL (Ravindran and Barto 2002; Abel, Hershkowitz, and Littman 2016; Abel et al. 2020), which are easily obtained together with the abstract MDP. Unlike other works, instead, we do not require any mapping between the action spaces. This relationship will remain implicit. This feature leaves high flexibility to the designers for defining the abstraction hierarchy.

An abstract model is therefore a suitable relaxation of the environment dynamics. For example, in a navigation scenario, an abstraction could contain actions that allow to just "leave the room", instead of navigating though space with low-level controls. In Section 5, we formalize this intuition

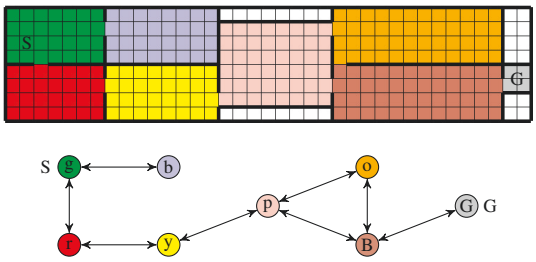


Figure 1: A grid world domain (top) and an abstraction (bottom). The colors encode the mapping function, G is the goal.

by deriving a measure that quantifies how accurate an abstraction is with respect to the lower domain.

As a simple example, consider the grid world domain, the abstract MDP and the mapping in Figure 1. Thanks to the abstraction, we can inform the agent that exploration should avoid the blue "room" (b) and only learn options for moving to and within the other rooms. The same does not necessarily hold for the orange block (o), instead, as the optimal path depends on the specific transition probabilities in each arc.

Exploiting the Knowledge

Let us consider a hierarchy of abstractions $\mathcal{M}_0, \ldots, \mathcal{M}_n$, together with the functions $\phi_0, \ldots, \phi_{n-1}$. The learning process proceeds incrementally, training in order from the easiest to the hardest model. When learning on $\mathcal{M}_i$, our method exploits the knowledge acquired from its abstraction by applying of a form of Reward Shaping, constructed from the estimated solution for $\mathcal{M}_{i+1}$. In particular, we recognize that the optimal value function V^*_{i+1} of $\mathcal{M}_{i+1}$ is a helpful heuristic that can be used to evaluate how desirable a group of states is according to the abstraction. We formalize our application of RS in the following definition.

Definition 2. Let $\mathcal{M}_i$ be an MDP and $\langle \mathcal{M}_{i+1}, \phi_i \rangle$ its abstraction. We define the *biased MDP* of $\mathcal{M}_i$ with respect to $\langle \mathcal{M}_{i+1}, \phi_i \rangle$ as the model $\mathcal{M}^b_i$, resulting from the application of reward shaping to $\mathcal{M}_i$, using the potential:

$$\Phi(s) \coloneqq V^*_{i+1}(\phi_i(s)) \tag{3}$$

where, V^*_{i+1} is the optimal value function of $\mathcal{M}_{i+1}$.

This choice is a novel contribution of this paper and it allows to evaluate each state according to how much desirable the corresponding abstract state is, according to the optimal policy for $\mathcal{M}_{i+1}$. This is beneficial, as high potentials are associated to high Q-function value initializations (Wiewiora 2003). In fact, (Ng, Harada, and Russell 1999) was the first to notice that the MDP own optimal value function is a very natural potential for RS. We extend this idea, by using the value function of the abstract MDP instead.

4 Reward Shaping for Episodic RL

Potential-Based RS has been explicitly designed not to alter the optimal policies. In fact, regardless of the potential, in case of an infinite horizon, or if the episodes always terminate in a zero-potential absorbing state, this is always guaranteed (Ng, Harada, and Russell 1999). However, in RL, it is extremely common to diversify the agent experiences by breaking up exploration in episodes of finite length. Thus, in the episodic setting, these guarantees do not hold anymore, as the episodes might terminate in states with arbitrary potential and the optimal policy can be altered (Grzes 2017).

To see this, consider an episode $\pi = s_0 a_1 r_1 s_1 \ldots s_n$ of an MDP $\mathcal{M}$, and the associated episode $\pi' = s_0 a_1 r'_1 s_1 \ldots s_n$, where rewards have been modified via reward shaping. The returns of the two sequences are related by (Grzes 2017):

$$G(\pi') \coloneqq \sum_{t=0}^{n-1} \gamma^t \, r'_{t+1} = G(\pi) + \gamma^n \, \Phi(s_n) - \Phi(s_0) \tag{4}$$

The term $\gamma^n \, \Phi(s_n)$ is the one responsible of modifying the optimal policies, as it depends on the state that is reached at the end of the episode. So, the solution proposed by (Grzes 2017) is to assume, for every terminal state, the null potential $\Phi(s_n) = 0$, as this would preserve the original returns.

However, this is not always the only desirable solution. In fact, we might be interested in relaxing the guarantee of an identical policy, in favour of a stronger impact on learning speed. The same need has been also identified by (Schubert, Oguz, and Toussaint 2021) and addressed with different solutions. As an example, let us consider an MDP with a null reward function everywhere, except when transitioning to a distinct goal state. As a consequence of equation (4) and the choice $\Phi(s_n) = 0$, all finite trajectories which do not contain the goal state are associated to the same return. Since the agent cannot estimate its distance to the goal through differences in return, return-invariant RS of (Grzes 2017) does not provide a *persistent* exploration bias to the agent. The form of RS adopted in this paper, which is formulated in Definition 2, does not assign null potentials to terminal states. Therefore, we say that it is *not* return-invariant. This explains why the MDP of Definition 2 has been called "biased": optimal policies of $\mathcal{M}^b_i$ and $\mathcal{M}_i$ do not necessarily correspond. This is addressed in the next section, where we show that the complete procedure recovers optimal convergence.

The Algorithm

Since we deliberately adopt a form of RS which is not return invariant in the episodic setting, we devised a technique to recover optimality in the original MDP, when coupled with any off-policy algorithm. The procedure is presented in detail in Algorithm 1. Learning proceeds sequentially, training from the most abstract model to the ground domain. When learning on the i-th MDP, the estimated optimal value function $\hat{V}^*_{i+1}$ of the previous model is used to obtain a reward shaping function (line 4). This implicitly defines the biased MDP $\mathcal{M}^b_i$ of Definition 2. Experience is collected by sampling actions according to a stochastic exploration policy, as determined by the specific learning algorithm $\mathcal{L}$. Such policy may be derived from the current optimal policy estimate for $\mathcal{M}^b_i$, such as an ϵ-greedy exploration policy in $\hat{\rho}^{b*}_i$ (line 9). Finally, the output of each learning phase are the estimates $\hat{\rho}^*_i, \hat{V}^*_i$ for the original MDP $\mathcal{M}_i$. This allows to iterate the process with an unbiased value estimate, or conclude the procedure with the final learning objective $\hat{\rho}^*_0$.

Algorithm 1: Main algorithm

Input: Off-policy learning algorithm $\mathcal{L}$
Input: $\mathcal{M}_0, \ldots, \mathcal{M}_n$, $\phi_0, \ldots, \phi_{n-1}$
Output: $\hat{\rho}_0^*$, ground MDP estimated optimum

1 $\hat{V}_{n+1}^* : s \mapsto 0$
2 $\phi_n : s \mapsto s$
3 **foreach** $i \in \{n, \ldots, 0\}$ **do**
4 $\quad F_i \leftarrow \textsc{Shaping}(\gamma_i, \phi_i, \hat{V}_{i+1}^*)$
5 $\quad \textit{Learner}_i \leftarrow \mathcal{L}(\mathcal{M}_i)$
6 $\quad \textit{Learner}_i^b \leftarrow \mathcal{L}(\mathcal{M}_i)$
7 $\quad$ **while** *not* $\textit{Learner}_i.\textsc{Stop}()$ **do**
8 $\quad\quad s \leftarrow \mathcal{M}_i.\textsc{State}()$
9 $\quad\quad a \leftarrow \textit{Learner}_i^b.\textsc{Action}(s)$
10 $\quad\quad r, s' \leftarrow \mathcal{M}_i.\textsc{Act}(a)$
11 $\quad\quad r^b \leftarrow r + F_i(s, a, s')$
12 $\quad\quad \textit{Learner}_i^b.\textsc{Update}(s, a, r^b, s')$
13 $\quad\quad \textit{Learner}_i.\textsc{Update}(s, a, r, s')$
14 $\quad$ **end**
15 $\quad \hat{\rho}_i^* \leftarrow \textit{Learner}_i.\textsc{Output}()$
16 $\quad \hat{V}_i^* \leftarrow \textsc{ComputeValue}(\hat{\rho}_i^*, \mathcal{M}_i)$
17 **end**

Proposition 1. *Let us consider MDPs $\mathcal{M}_0, \ldots, \mathcal{M}_n$ and their associated mapping functions $\phi_0, \ldots, \phi_{n-1}$. If $\mathcal{L}$ is an off-policy learning algorithm, then, in every i-th iteration of Algorithm 1, $\hat{\rho}_i^*$ converges to ρ_i^*, as the number of environment interactions increase.*

Proof. All proofs are in the appendix[1]. □

5 Abstraction Quality

Our approach gives great flexibility in selecting an abstraction. Still, given some ground MDP, not all models are equally helpful as abstractions. This section serves to define what properties should good abstractions possess. As we can see from Algorithm 1, they are used to construct effective exploration policies (row 9). Therefore, they should induce a biased MDP that assigns higher rewards to regions of the state space from which the optimal policy of the original problem can be easily estimated. The exploration loss of an abstraction, introduced in Definition 4, captures this idea.

Although we may apply the proposed method in generic MDPs, our analysis focuses on a wide class of tasks that can be captured with goal MDPs.

Definition 3. We say that an MDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, T, R, \gamma \rangle$ is a *goal MDP* iff there exists a set of goal states $\mathcal{G} \subseteq \mathcal{S}$ such that:

$$R(s, a, s') = 1 \quad \text{if } s \notin \mathcal{G} \text{ and } s' \in \mathcal{G}, \quad 0 \text{ otherwise} \tag{5}$$

$$V^*(s) = 0 \qquad \forall s \in \mathcal{G} \tag{6}$$

[1]The extended version is at arXiv:2303.00516 [cs.LG].

Equation (6) simply requires that from any goal state, it is not possible to re-enter any other goal and collect an additional reward. Goal MDPs are very straightforward definitions of tasks, which are also sufficiently general, as we see in the experimental section.

Assumption 1. The ground MDP $\mathcal{M}_0$ is a goal MDP.

Assumption 2. Given a goal MDP $\mathcal{M}_i$, with goal states $\mathcal{G}_i$, and its abstraction $\langle \mathcal{M}_{i+1}, \phi_i \rangle$, we assume $\mathcal{M}_{i+1}$ is a goal MDP with $\mathcal{G}_{i+1}$ satisfying:

$$\mathcal{G}_i = \cup_{s \in \mathcal{G}_{i+1}} \phi^{-1}(s) \tag{7}$$

In other words, abstract goals should correspond through the mapping to all and only the goal states in the ground domain. In the example of Figure 1, the gray cells in the ground MDP are mapped to the abstract goal labelled as G.

We start our analysis with two observations. First, due to how the framework is designed, convergence on any model $\mathcal{M}_i$, does depend on its abstraction, $\mathcal{M}_{i+1}$, but not on any other model in the hierarchy. Therefore, when discussing convergence properties, it suffices to talk about a generic MDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, T, R, \gamma \rangle$ and its direct abstraction, $\bar{\mathcal{M}} = \langle \bar{\mathcal{S}}, \bar{\mathcal{A}}, \bar{T}, \bar{R}, \bar{\gamma} \rangle$. Let $\phi : \mathcal{S} \to \bar{\mathcal{S}}$ denote the relevant mapping. Second, while a goal MDP $\mathcal{M}$ has sparse rewards, the reward function of the biased MDP $\mathcal{M}^b$ is no longer sparse. Depending on the abstraction $\langle \bar{\mathcal{M}}, \phi \rangle$, from which it is defined, the rewards of $\mathcal{M}^b$ can be as dense as needed. As confirmed empirically, this allows to achieve a faster convergence on the biased MDP. However, its optimum ρ^{b*} should also be a good exploration policy for the original domain. Therefore, we measure how similar ρ^{b*}, the optimal policy for the biased MDP $\mathcal{M}^b$, is with respect to some optimal policy ρ^* of $\mathcal{M}$.

Definition 4. Given an MDP $\mathcal{M}$, the *exploration loss* of an abstraction $\langle \bar{\mathcal{M}}, \phi \rangle$ is the expected value loss of executing in $\mathcal{M}$ the optimal policy of $\mathcal{M}^b$, ρ^{b*}, instead of its optimal policy:

$$L(\mathcal{M}, \langle \bar{\mathcal{M}}, \phi \rangle) \coloneqq \max_{s \in \mathcal{S}} |V^*(s) - Q(s, \rho^{b*})| \tag{8}$$

In order to limit this quantity, we relate abstract states $\bar{s} \in \bar{\mathcal{S}}$ to sets of states $\phi^{-1}(\bar{s}) \subseteq \mathcal{S}$ in the ground MDP. Similarly, actions $\bar{a} \in \bar{\mathcal{A}}$ correspond to non-interruptible policies that only terminate when leaving the current block. So, a more appropriate correspondence can be identified between abstract actions and ϕ-relative options in $\mathcal{M}$. We start by deriving, in equation (9), the multi-step value of a ϕ-relative option in goal MDPs.

Multi-step value of options By combining the classic multi-step return of options (Sutton, Precup, and Singh 1999), ϕ-relative options from (Abel et al. 2020) and goal MDPs of Definition 3, we obtain:

Lemma 1. *Given a goal MDP $\mathcal{M}$ and $\phi : \mathcal{S} \to \bar{\mathcal{S}}$, for any $s \in \mathcal{S}$ and ϕ-relative option o, the optimal value of o is:*

$$Q^*(s, o) = \sum_{k=0}^{\infty} \gamma^k \sum_{s_{1:k} \in \phi(s)^k} \sum_{s' \notin \phi(s)} \Big(p(s_{1:k} s' \mid s, \rho_o) \left(\mathbb{I}(s' \in G) + \gamma V^*(s') \right) \Big) \tag{9}$$

This expression sums over any sequence of states $s_1 \dots s_k$ remaining within $\phi(s)$ and leaving the block after k steps at s'. A similar result was derived by (Abel et al. 2020) for a slightly different definition for goal MDPs. However, equation (9) is not an expression about abstract states, yet, because it depends on the specific ground state s' that is reached at the end of the option. Therefore, in the following definition, we introduce a parameter ν that quantifies how much the reachable states s' in each block are dissimilar in value. This allows to jointly talk about the value of each group of states as a whole. We define a function $W_\nu : \bar{\mathcal{S}} \times \bar{\mathcal{S}} \to \mathbb{R}$, that, given a pair of abstract states $\bar{s}, \bar{s}'$, predicts, with ν-approximation error, the value of any successor ground state $s' \in \bar{s}'$ that can be reached from some $s \in \bar{s}$.

Definition 5. Consider an MDP $\mathcal{M}$ and an abstraction $\langle \bar{\mathcal{M}}, \phi \rangle$. We define the *abstract value approximation* as the smallest $\nu \geq 0$ such that there exists a function $W_\nu : \bar{\mathcal{S}} \times \bar{\mathcal{S}} \to \mathbb{R}$ which, for all $\bar{s}, \bar{s}' \in \bar{\mathcal{S}}$, satisfies:

$$\forall s \in \bar{s}, \forall s' \in \bar{s}', \forall a \in \mathcal{A} \\ T(s,a,s') > 0 \Rightarrow |W_\nu(\bar{s}, \bar{s}') - V^*(s')| \leq \nu \quad (10)$$

According to this definition, the frontier separating any two sets of states in the partition induced by ϕ must lie in ground states that can be approximated with the same optimal value, with a maximum error ν. Thus, any small ν puts a constraint on the mapping function ϕ. In the example of Figure 1, each room is connected to each other though a single location, so this condition is simply satisfied for $\nu = 0$. However, this definition allows to apply our results in the general case, provided that the ground states between two neighboring regions can be approximated to be equally close to the goal, with a maximum error of ν.

Thanks to Definition 5, it is possible to bound the value of options, only taking future abstract states into consideration (Lemma 2). For this purpose, when starting from some $s \in \mathcal{S}$, we use $p(\bar{s}', k \mid s)$ to denote the probability of the event of remaining for k steps in the same block as s, then reaching any $s' \in \bar{s}'$ at the next transition.

Lemma 2. *Let $\mathcal{M}$ be an MDP and $\langle \bar{\mathcal{M}}, \phi \rangle$ its abstraction, satisfying assumptions 1 and 2. The value of any ϕ-relative option o in $\mathcal{M}$ admits the following lower bound:*

$$Q^*(s,o) \geq \sum_{\bar{s}' \in \bar{\mathcal{S}} \setminus \{\phi(s)\}} \sum_{k=0}^{\infty} \gamma^k \, p(\bar{s}', k \mid s, \rho_o) \, \big(\\ \mathbb{I}(\bar{s}' \in \bar{\mathcal{G}}) + \gamma \, (W_\nu(\phi(s), \bar{s}') - \nu) \big) \quad (11)$$

at any $s \in \mathcal{S}$, where, ν and W_ν follow Definition 5.

This lemma provides a different characterization of options, in terms of abstract states, so that it can be exploited to obtain Theorem 1.

Exploration loss of abstractions Thanks to the results of the previous section, we can now provide a bound for the exploration loss of Definition 4, for any abstraction. We expand the results from (Abel et al. 2020) to limit this quantity.

First, we observe that any policy can be regarded as a finite set of ϕ-relative options whose initiation sets are partitioned according to ϕ. Then, from Lemma 2, we know that an approximation for the value of options only depends on the k-step transition probability to each abstract state. So, we assume this quantity is bounded by some ϵ:

Definition 6. Given an MDP $\mathcal{M}$, a function $\phi : \mathcal{S} \to \bar{\mathcal{S}}$ and two policies ρ_1, ρ_2, we say that ρ_1 and ρ_2 have *abstract similarity* ϵ if

$$\forall s \in \mathcal{S}, \quad \forall \bar{s}' \in \bar{\mathcal{S}} \setminus \{\phi(s)\}, \quad \forall k \in \mathbb{N} \\ | \, p(\bar{s}', k \mid s, \rho_1) - p(\bar{s}', k \mid s, \rho_2) \, | \leq \epsilon \quad (12)$$

Intuitively, abstract similarity measures the difference between the two abstract actions described by each policy, as it only depends on the probability of the next abstract state that is reached, regardless of the single trajectories and the specific final ground state. In the running example of Figure 1, two policies with low ϵ, after the same number of steps, would reach the same adjacent room with similar probability. It is now finally possible to state our result.

Theorem 1. *Let $\mathcal{M}$ and $\langle \bar{\mathcal{M}}, \phi \rangle$ be and MDP and an abstraction satisfying assumptions 1 and 2, and let $\mathcal{M}^b$ be the biased MDP. If ϵ is the abstract similarity of ρ^* and ρ^{b*}, and the abstract value approximation is ν, then, the exploration loss of $\langle \bar{\mathcal{M}}, \phi \rangle$ satisfies:*

$$L(\mathcal{M}, \langle \bar{\mathcal{M}}, \phi \rangle) \leq \frac{2|\bar{\mathcal{S}}|(\epsilon + \gamma \, \nu)}{(1-\gamma)^2} \quad (13)$$

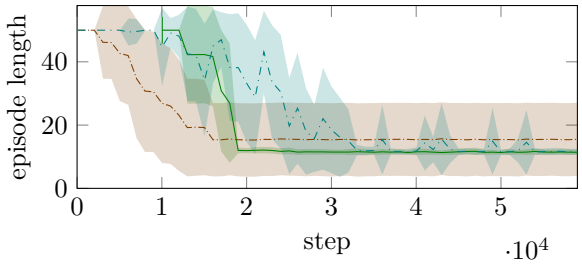
This theorem shows under which conditions an abstraction induces an exploration policy that is similar to some optimal policy of the original domain. However, we recall that optimal convergence is guaranteed regardless of the abstraction quality, because the stochastic exploration policy satisfies the mild conditions posed by the off-policy learning algorithm adopted.

Apart from our application to the biased MDP policy, the theorem has also a more general impact. The result shows that, provided that the abstraction induces a partition of states whose frontiers have some homogeneity in value (Definition 5), it is possible to reason in terms of abstract transitions. Only for a $\nu = 0$, this bound has similarities with the inequality n. 5 in (Abel et al. 2020). Notice however, that the one stated here is expressed in terms of the size of the abstract state space, which usually can be assumed to be $|\bar{\mathcal{S}}| \ll |\mathcal{S}|$.

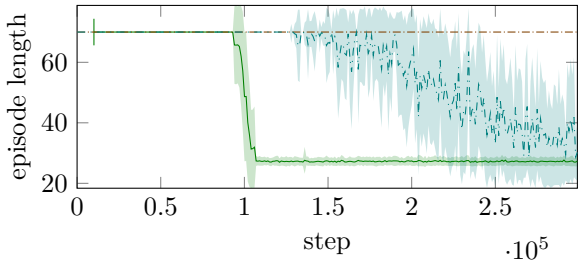
6 Validation

We initially consider a navigation scenario, where some locations in a map are selected as goal states.

Environments We start with two levels of abstractions, $\mathcal{M}_1$ and $\mathcal{M}_2$. The ground MDP $\mathcal{M}_1$ consists of a finite state space $\mathcal{S}_1$, containing a set of locations, and a finite set of actions $\mathcal{A}_1$ that allows to move between neighboring states,



(a) Navigation task in the 4-rooms domain.



(b) Navigation task in the 8-rooms domain.

Figure 2: Results on the navigation tasks.

with some small failure probability. Following the idea of Figure 1, we also define an abstract MDP $\mathcal{M}_2$, whose states correspond to contiguous regions of $\mathcal{S}_1$. Actions $\mathcal{A}_2$ allow to move, with high probability, from any region to any other, only if there is a direct connection in $\mathcal{M}_1$. We instantiate this idea in two domains. In the first, we consider a map as the one in the classic 4-rooms environment from (Sutton, Precup, and Singh 1999). The second, is the "8-rooms" environment shown in Figure 1.[2]

Training results In the plots of Figures 2a and 2b, for each of the two ground MDPs, we compare the performance of the following algorithms:

- Q-learning (Watkins and Dayan 1992);
- Delayed Q-learning (Strehl et al. 2006);
- Algorithm 1 (our approach) with Q-learning.

Each episode is terminated after a fixed timeout or when the agent reaches a goal state. Therefore, lower episode lengths are associated to higher cumulative returns. Horizontal axis spans the number of sampled transitions. Each point in these plots shows the average and standard deviation of the evaluations of 10 different runs. The solid green line of our approach is shifted to the right, so to account for the number of time steps that were spent in training the abstraction. Further training details can be found in the appendix. As we can see from Figure 2a, all algorithms converge relatively easily in the smaller 4-rooms domain. In Figure 2b, as the state space increases and it becomes harder to explore, a naive exploration policy does not allow Q-learning to converge in reasonable time. Our agent, on the other hand, steadily converge to optimum, even faster than Delayed Q-learning which has polynomial time guarantees.

[2]Code available at https://github.com/cipollone/multinav2

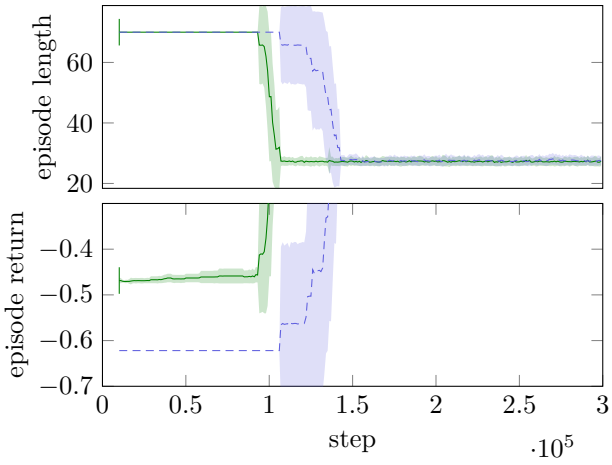


Figure 3: Return-invariant RS and our approach.

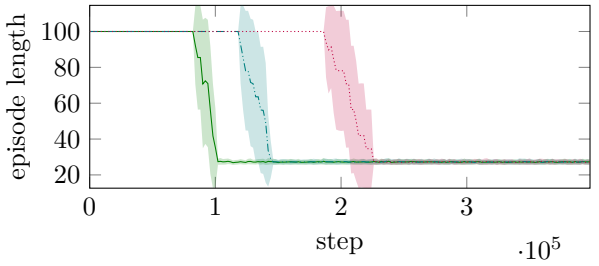


Figure 4: Training in presence of errors.

Return-Invariant Shaping

As discussed in Section 4, when applying RS in the episodic setting, there is a technical but delicate distinction to make between:

- Return-invariant RS (null potentials at terminal states);
- Non return-invariant RS (our approach).

In Figure 3 (top), we compare the two variants on the 8-rooms domain. Although both agents receive RS from the same potential, this minor modification suffices to produce this noticeable difference. The reason lies in the returns the two agents observe (bottom). Although they are incomparable in magnitude, in the early learning phase, we see that only our reward shaping is able to reward each episode differently, depending on their estimated distance to the goal.

Robustness to Modelling Errors

We also considered the effect of significant modelling errors in the abstraction. In Figure 4, we report the performance of our agent on the 8-rooms domain, when driven by three different abstractions:

- $\mathcal{M}_2$: is the same abstraction used in Figure 2b;
- $\mathcal{M}_2^{(b)}$: is $\mathcal{M}_2$ with an additional transition from the pink states (p) to the goal (G), not achievable in $\mathcal{M}_1$.
- $\mathcal{M}_2^{(c)}$: is $\mathcal{M}_2^{(b)}$ with an additional transition from the blue (b) to the pink region (p), not achievable in $\mathcal{M}_1$.

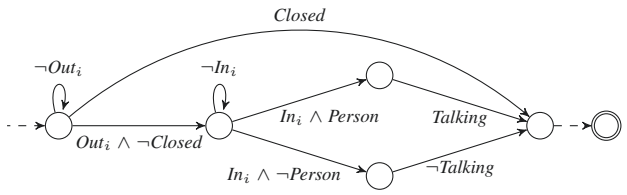


Figure 5: A temporally-extended task, repeated for $i = 1, 2$. The missing transitions go to a failure sink state.

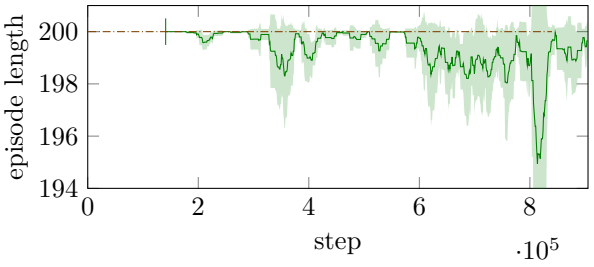


Figure 6: Dueling DQN algorithm with and without our RS. Training episode lengths, averaged over 5 runs.

Clearly, abstractions with bigger differences with respect to the underlying domain cause the learning process to slow down. However, with any of these, Q-learning converges to the desired policy and the performance degrades gracefully. Interestingly, even in presence of severe modelling errors, the abstraction still provides useful information with respect to uninformed exploration.

Interaction Task

In this section, we demonstrate that the proposed method applies to a wide range of algorithms, dynamics and tasks. With respect to variability in tasks, we emphasize that goal MDPs can capture many interesting problems. For this purpose, instead of reaching a location, we consider a complex temporally-extended behavior such as: "reach the entrance of the two rooms in sequence and, if each door is open, enter and interact with the person inside, if present". This task is summarized by the deterministic automaton $\mathcal{D}$ of Figure 5. Note that there is a single accepting state, and arcs are associated to environment events.

Regarding the environment dynamics, instead, we define $\mathcal{M}_{2,d}$ and $\mathcal{M}_{1,d}$, respectively the abstract and grid transition dynamics seen so far. In addition, we consider a ground MDP $\mathcal{M}_{0,d}$ at which the robot movements are modelled using continuous features. The state space $\mathcal{S}_0$ now contains continuous vectors (x, y, θ, v), representing pose and velocity of agent's mobile base on the plane. The discrete set of actions $\mathcal{A}_0$ allows to accelerate, decelerate, rotate, and a special action denotes the initiation of an interaction.

There exists goal MDPs, $\mathcal{M}_2, \mathcal{M}_1, \mathcal{M}_0$ that capture both the dynamics and the task defined above, which can be obtained through a suitable composition of each $\mathcal{M}_{i,d}$ and $\mathcal{D}$ (Brafman, De Giacomo, and Patrizi 2018; Icarte et al. 2018). Therefore, we can still apply our tecnique to the composed goal MDP. Since $\mathcal{M}_0$ now includes continuous features we adopt Dueling DQN (Wang et al. 2016), a Deep RL algorithm. The plot in Figure 6 shows a training comparison between the Dueling DQN agent alone (dot-dashed brown), and Dueling DQN receiving rewards from the grid abstraction (green). As we can see, our method allows to provide useful exploration bias even in case of extremely sparse goal states, as in this case.

7 Related Work

Hierarchical RL specifically studies efficiency in presence of abstractions. Some classic approaches are MAXQ (Dietterich 2000), HAM (Parr and Russell 1998) and options (Sutton, Precup, and Singh 1999). Instead of augmenting the ground MDP with options, which would result in a semi-MDP, we use them as formalizations of partial policies. In order to describe which relation should the ground MDP and its abstraction satisfy, (Ravindran and Barto 2002; Li, Walsh, and Littman 2006) develop MDP Homomorphisms and approximated extensions. Differently to these works, our method does not try to capture spatial regularities in the domain, rather, we are interested in coarse partitioning of neighboring states, which can hardly be approximated with a single value. This issue also appeared in (Jothimurugan, Bastani, and Alur 2021) in the form of non-Markovianity.

Our abstractions are closely related to those described in (Abel, Hershkowitz, and Littman 2016; Abel et al. 2020), in which both the state and the action spaces differ. Still, they do not exploit, as in our work, explicit abstract MDPs, since they only learn in one ground model. Regarding the use of Reward Shaping, (Gao and Toni 2015) presented the idea of applying RS in context of HRL, and applying it specifically to the MAXQ algorithm. Recently, (Schubert, Oguz, and Toussaint 2021) proposed a new form of biased RS for goal MDPs, with looser convergence guarantees. With a different objective with respect to this paper, various works consider how abstractions may be learnt instead of being pre-defined, including (Marthi 2007; Grzes and Kudenko 2008; Steccanella, Totaro, and Jonsson 2021). This is an interesting direction to follow and the models that are obtained with such techniques may be still exploited with our method.

8 Conclusion

In this paper, we have presented an approach to increase the sample efficiency of RL algorithms, based on a linear hierarchy of abstract simulators and a new form of reward shaping. While the ground MDP accurately captures the environment dynamics, higher-level models represent increasingly coarser abstractions of it. We have described the properties of our RS method under different abstractions and we have shown its effectiveness in practice. Importantly, our approach is very general, as it makes no assumptions on the off-policy algorithm that is used, and it has minimal requirements in terms of mapping between the abstraction layers. As future work, we plan to compare our technique with other methods from Hierarchical RL, especially those with similar prior assumptions and to evaluate them in a robotic application with low-level perception and control.

Acknowledgements

This work has been supported by the ERC Advanced Grant WhiteMech (No. 834228), by the EU ICT-48 2020 project TAILOR (No. 952215), and by the PRIN project RIPER (No. 20203FFYLK).

References

Abel, D.; Hershkowitz, D.; and Littman, M. 2016. Near Optimal Behavior via Approximate State Abstraction. In *ICML 2016*, volume 48, 2915–2923. PMLR.

Abel, D.; Umbanhowar, N.; Khetarpal, K.; Arumugam, D.; Precup, D.; and Littman, M. 2020. Value Preserving State-Action Abstractions. In *AISTATS*, volume 108, 1639–1650. PMLR.

Brafman, R. I.; De Giacomo, G.; and Patrizi, F. 2018. LTLf / LDLf Non-Markovian Rewards. *AAAI 2018*, 1771–1778.

Devlin, S.; and Kudenko, D. 2012. Dynamic Potential-Based Reward Shaping. In *AAMAS 2012*, 433–440. IFAAMAS.

Dietterich, T. G. 2000. Hierarchical reinforcement learning with the MAXQ value function decomposition. *JAIR*, 13: 227–303.

Gao, Y.; and Toni, F. 2015. Potential Based Reward Shaping for Hierarchical Reinforcement Learning. In *IJCAI 2015*, 3504–3510. AAAI Press.

Grzes, M. 2017. Reward Shaping in Episodic Reinforcement Learning. In *AAMAS 2017*, 565–573. ACM.

Grzes, M.; and Kudenko, D. 2008. Multigrid Reinforcement Learning with Reward Shaping. In *ICANN 2008, Part I*, volume 5163, 357–366. Springer.

Hutsebaut-Buysse, M.; Mets, K.; and Latré, S. 2022. Hierarchical Reinforcement Learning: A Survey and Open Research Challenges. *Machine Learning and Knowledge Extraction*, 4(1): 172–221.

Icarte, R. T.; Klassen, T.; Valenzano, R.; and McIlraith, S. 2018. Using reward machines for high-level task specification and decomposition in reinforcement learning. In *ICML*, volume 80, 2107–2116. PMLR.

Jothimurugan, K.; Bastani, O.; and Alur, R. 2021. Abstract Value Iteration for Hierarchical Reinforcement Learning. In *AISTATS*, volume 130, 1162–1170. PMLR.

Li, L.; Walsh, T. J.; and Littman, M. L. 2006. Towards a Unified Theory of State Abstraction for MDPs. In *ISAIM 2006*, 531–539.

Marthi, B. 2007. Automatic shaping and decomposition of reward functions. In *ICML 2007*, volume 227, 601–608. ACM.

Ng, A. Y.; Harada, D.; and Russell, S. J. 1999. Policy Invariance Under Reward Transformations: Theory and Application to Reward Shaping. In *ICML 1999*, 278–287. Morgan Kaufmann.

Parr, R.; and Russell, S. 1998. Reinforcement learning with hierarchies of machines. *Advances in neural information processing systems*, 1043–1049.

Puterman, M. L. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley. ISBN 978-0-471-61977-2.

Ravindran, B.; and Barto, A. G. 2002. Model Minimization in Hierarchical Reinforcement Learning. In *SARA 2002*, volume 2371, 196–211. Springer.

Schubert, I.; Oguz, O. S.; and Toussaint, M. 2021. Plan-Based Relaxed Reward Shaping for Goal-Directed Tasks. In *ICLR 2021*. OpenReview.net.

Steccanella, L.; Totaro, S.; and Jonsson, A. 2021. Hierarchical Representation Learning for Markov Decision Processes. *CoRR*, abs/2106.01655.

Strehl, A. L.; Li, L.; Wiewiora, E.; Langford, J.; and Littman, M. L. 2006. PAC model-free reinforcement learning. In *ICML 2006*, 881–888.

Sutton, R. S.; Precup, D.; and Singh, S. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artif. Intell.*, 112(1-2): 181–211.

Wang, Z.; Schaul, T.; Hessel, M.; van Hasselt, H.; Lanctot, M.; and de Freitas, N. 2016. Dueling Network Architectures for Deep Reinforcement Learning. In *ICML*, volume 48, 1995–2003. JMLR.org.

Watkins, C. J.; and Dayan, P. 1992. Q-learning. *Machine learning*, 8(3-4): 279–292.

Wiewiora, E. 2003. Potential-Based Shaping and Q-Value Initialization Are Equivalent. *Journal of Artificial Intelligence Research*, 19: 205–208.

Wiewiora, E.; Cottrell, G. W.; and Elkan, C. 2003. Principled Methods for Advising Reinforcement Learning Agents. In *ICML 2003*, 792–799. AAAI Press.