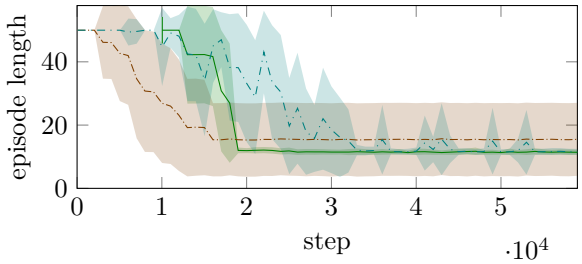
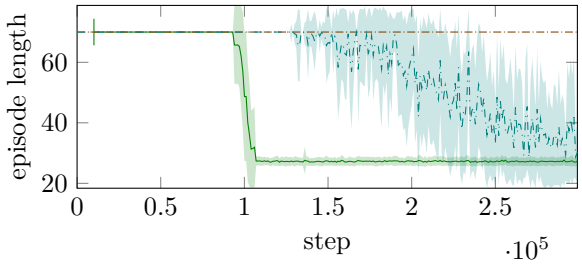


(a) Navigation task in the 4-rooms domain.



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

Figure 2: Results on the navigation tasks.

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 (**?**). The second, is the "8-rooms" environment shown in Figure 1.[1]

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 (**?**);
- Delayed Q-learning (**?**);
- 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.

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

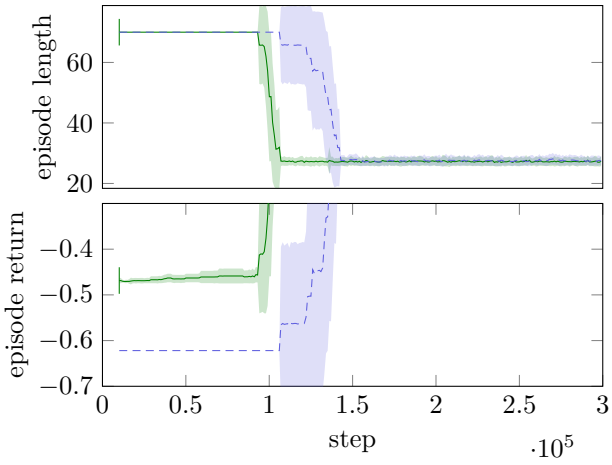


Figure 3: Return-invariant RS and our approach.

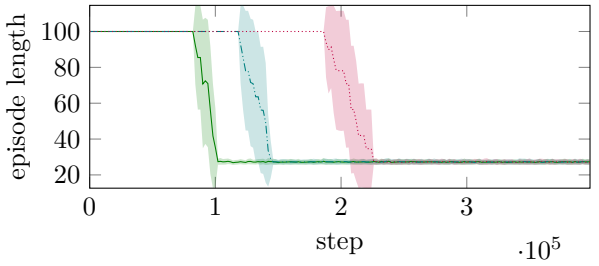


Figure 4: Training in presence of errors.

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 $\mathcal{M}_1$.

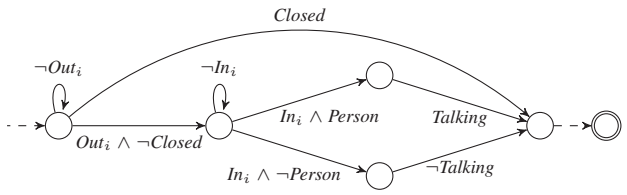


Figure 5: A temporally-extended task, repeated for $i = 1, 2$. The missing transitions go to a failure sink state.

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 $\mathcal{M}_1, \mathcal{M}_2, \ldots \mathcal{M}_T$ 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 $\mathcal{D}$ (**??**). Therefore, we can still apply our tecnique to the composed goal MDP. Since $\mathcal{M}_0$ now includes continuous features we adopt Dueling DQN (**?**), 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 (**?**), HAM (**?**) and options (**?**). 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, (**??**) develop MDP Homomorphisms and

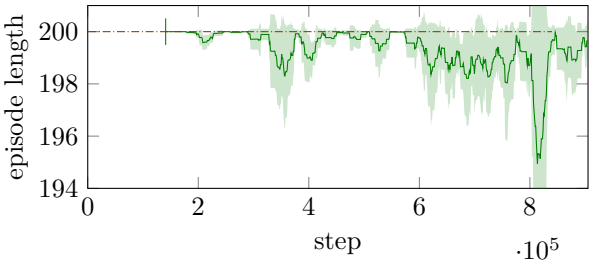


Figure 6: Dueling DQN algorithm with and without our RS. Training episode lengths, averaged over 5 runs.

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 (**?**) in the form of non-Markovianity. Our abstractions are closely related to those described in (**??**), 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, (**?**) presented the idea of applying RS in context of HRL, and applying it specifically to the MAXQ algorithm. Recently, (**?**) 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 (**???**). 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).

Totam esse optio accusamus inventore odio, voluptates ipsum aperiam earum blanditiis deleniti aut quis rem ad necessitatibus, dolor dolorem ipsa amet laudantium esse