Online Learning a Symbolic Abstraction of Actions in Hierarchical RL with Formal Methods

Sao Mai Nguyen[a;*]

[a]Flowers Team, U2IS, ENSTA Paris, IP Paris
& IMT Atlantique, Lab-STICC, UMR CNRS 6285
ORCiD ID: Sao Mai Nguyen https://orcid.org/0000-0003-0929-0019

Abstract. Reasoning about actions, planning in open-ended learning requires a hierarchical abstractions of action models and benefits from the use of symbolic methods for goal representation as they structure knowledge for efficient and transferable learning. However, the existing Hierarchical Reinforcement Learning (HRL) approaches relying on symbolic reasoning are often limited as they require a manual goal representation. A symbolic goal representation must preserve information about the environment dynamics. We propose an automatic subgoal discovery via an emergent representation that abstracts (i.e., groups together) sets of environment states that have similar roles in the task, using formal verification methods. We create a HRL algorithm that learns online this representation along with the policies. We show on navigation tasks that the learned representation is interpretable and results in data efficiency.

1 Introduction

As robots stepping out of factories need to learn in open-ended high-dimensional sensorimotor spaces, symbol emergence is key for allowing symbolic reasoning, compositionality, hierarchical organisation of the knowledge, etc. Sensorimotor symbol emergence thus is key to scaling up primitive actions into complex actions for open-ended learning, using compositionality [6] and hierarchy [2].

Action hierarchies are the core idea of Hierarchical Reinforcement Learning (HRL) that decomposes a task into easier subtasks. In particular, in Feudal HRL [1] a high-level agent selects subgoals that a low-level agent learns to achieve. The performance of Feudal HRL depends on the "hierarchical division of the available state space" [1], the representation of the goals that the high level agent uses to decompose a task. Yet, only few algorithms learn it automatically [8], while others either use directly the state space [7] or manually provide a representation [4, 11]. In this research, we tackle the problem of learning automatically, while learning the policy, a discrete interpretable goal representation from continuous observations that expresses the task structure for data-efficiency.

We introduced in [9, 10] a novel goal space representation for a feudal HRL algorithm, a hierarchical abstraction of action models combining reinforcement learning and formal methods. In [10], we proposed a novel three-layer HRL algorithm, named Spatio-Temporal Abstraction via Reachability (STAR), that learns the goal space abstraction through reachability analysis. Thus, **STAR exploits an emergent discrete representation of the goal space, to be used more easily for task composition into sequential tasks**.

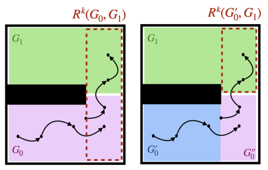


Figure 1: In this maze, the transition starting from a state in G_0 for k steps may reach both G_0 and G_1. G_0 is thus split into two regions where all the states from G_0' can reach G_1 and the states in G_0 don't.

2 Methodology

STAR uses as framework, a goal-conditioned Markov Decision Process $(\mathcal{S}, \mathcal{A}, P, r_{ext})$, where $\mathcal{S} \subseteq \mathbb{R}^n$ is a continuous state space, $\mathcal{A}$ is an action space, $P(s_{t+1}|s_t, a_t)$ is a probabilistic transition function, and r_{ext} is a parameterised reward function, defined as the negative distance to the task goal $g^* \in \mathcal{S}$, i.e $r_{\text{ext}}(s, g^*) = -\|g^* - s\|_2$. The *multi-task reinforcement learning problem* consists in learning a goal conditioned policy π to sample at each time step t an action $a \sim \pi(s_t \mid g^*)$, so as to maximize the expected cumulative reward. The spatial goal abstraction is modeled by a *set-based abstraction* defined by a function $\mathcal{N} : \mathcal{S} \to 2^{\mathcal{S}}$ that maps concrete states to sets of states (i.e., $\forall s \in \mathcal{S}, \mathcal{N}(s) \subseteq \mathcal{S}$). We write $\mathcal{G}_{\mathcal{N}}$ to refer to the range of the abstraction $\mathcal{N}$, which is intuitively the abstract goal space.

STAR learns, at the same time, a spatial goal abstraction $\mathcal{N}$ and policies at multiple time scales. The STAR algorithm, shown in Figure 2, has two main components: a 3-levels Feudal HRL algorithm (enclosed in the red dashed lines); and an abstraction refinement component (shown in the blue solid lines). STAR runs the Feudal HRL algorithm and the abstraction refinement in a feedback loop, refining the abstraction $\mathcal{N}$ at the end of every learning episode.

Our work presents the following contributions:

1. A novel Feudal HRL algorithm, STAR, to learn online (Fig. 2) with the three RL agents :
   (a) *Commander*: the highest-level agent learns the policy $\pi_{\text{Comm}} : \mathcal{S} \times \mathcal{S} \to \mathcal{G}$ that is a goal-conditioned on g^* and samples an abstract **goal** $G \in \mathcal{G}$ that should help to reach the task goal g^* from the current agent's state $(G_{t+k} \sim \pi_{\text{Comm}}(s_t, g^*))$.
   (b) *Tutor*: the mid-level agent is conditioned by the *Commander* goal G. It learns the policy $\pi_{\text{Tut}} : \mathcal{S} \times \mathcal{G} \to \mathcal{S}$ and picks **subgoals** in the state space $(g_{t+l} \sim \pi_{\text{Tut}}(s_t, G_{t+k}))$.

[*] Corresponding Author. Email: nguyensmai@gmail.com

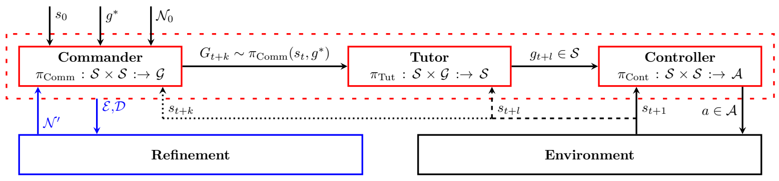

s_0
g^*
$\mathcal{N}_0$
Commander
$\pi_{\text{Comm}} : \mathcal{S} \times \mathcal{S} :\to \mathcal{G}$
$G_{t+k} \sim \pi_{\text{Comm}}(s_t, g^*)$
Tutor
$\pi_{\text{Tut}} : \mathcal{S} \times \mathcal{G} :\to \mathcal{S}$
$g_{t+l} \in \mathcal{S}$
Controller
$\pi_{\text{Cont}} : \mathcal{S} \times \mathcal{S} :\to \mathcal{A}$
$\mathcal{N}'$
$\mathcal{E},\mathcal{D}$
s_{t+k}
s_{t+l}
s_{t+1}
$a \in \mathcal{A}$
Refinement
Environment

Figure 2: **Architecture of STAR**. STAR's inputs are the initial state s_0, the task goal g^*, and an initial abstraction $\mathcal{N}_0$. STAR runs a Feudal HRL algorithm (dashed red block) and an abstraction refinement (blue box). The solid red blocks show the HRL agents (Commander, Tutor, Controller). The agents run at different timescales ($k > l > 1$), shown with the solid, dashed, and dotted lines carrying the feedback from the environment to the agents. The Refinement uses as inputs the past episodes ($\mathcal{D}$) and a the list of abstract goals ($\mathcal{E}$) visited during the last episode, and outputs an abstraction.

(c) *Controller*: the low-level policy $\pi_{\text{Cont}} : \mathcal{S} \times \mathcal{S} \to \mathcal{A}$ is goal-conditioned by the *Tutor*'s subgoal g and samples actions to reach given goal ($a \sim \pi_{\text{Cont}}(s_t, g_{t+l})$).

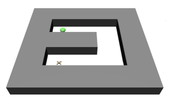
2. An emergent symbolic representation of the environment, based on reachability analysis. Each symbol representing a region of the state space, we estimate the k-step reachability of the forward model from each region. Technically, we compute an *over-approximation* of the image of the forward model with the Ai2 [3] tool. We refine the representation by splitting the regions online, so that the partition satisfies the pairwise reachability property (Fig. 1). We provide a theoretical motivation for using reachability-aware goal representations, showing convergence to a reachability-aware abstraction after applying a finite number of refinements, and showing a guarantee of a suboptimality bound for the converged policy. Compared to the optimal policy, the policy learned on the abstraction has a near optimal value : $|V_{\pi*}(s_i) - V_{\pi*_{\mathcal{N}}}| \leq U(\epsilon, i, \gamma)|$, where U is a function of γ the discount factor, ϵ a bound on reward values, and i the time step index.
3. Empirical results showing that STAR successfully combines both temporal and spatial abstraction for more efficient learning, and that the reachability-aware abstraction scales to tasks with more complex dynamics in Ant environments (Fig. 3).

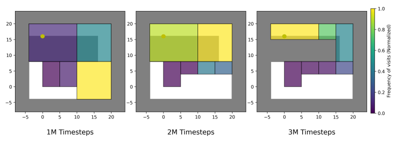
3 Experimental Results

Our evaluation in [10] shows superior success rate of STAR compared to the state of the art [11, 5, 7].

We examine the abstraction learned by STAR's *Commander* agent at different timesteps during learning when solving the Ant Maze (Fig. 3a). Fig. 3b shows how STAR gradually refines the goal abstraction to identify successful trajectories in the environment. Progressively, the ant explores trajectories leading to the goal of the task. Additionally, the frequency of visiting goals in the difficult areas of the maze (e.g., the tight corners) is higher, and these goals are eventually refined in the training, fitting with the configuration of the obstacles. STAR learns a more precise abstraction in bottleneck areas where only a few subset of states manage to reach the next goal.

We evaluate GARA [9] and examine how RL can combine with planning on the 2 Paths Maze configuration (Fig. 4a). In such configuration there are two ways to solve the maze: from the starting position the agent can either take the upper path, which is harder to train a low-level policy for, or take the narrow, harder to discover, shorter path at the bottom. The obtained subgoal representation can be used for planning : the graph in Fig. 4b shows the connected transitions in the learned partitions. Our hierarchical abstraction of action models can be used for both planning and RL.


(a) Ant Maze Environment



(b) Learned Representation

Figure 3: **Ant Maze:** in this task, the ant must navigate a '⊃'-shaped maze to reach the exit positioned at the top left. **Learned Abstraction:** Frequency of goals visited by the *Commander* when evaluating a policy learned after 1M, 2M, and 3M timesteps (averaged over 5 different evaluations with 500 maximum timesteps). The subdivision of the mazes represent (abstract) goals. The color gradient represents the frequency of visits of each goal; Grey areas, the obstacles.

References

[1] Peter Dayan and Geoffrey E Hinton, 'Feudal reinforcement learning', in *NeurIPS*, volume 5, (1992).

[2] Nicolas Duminy, Sao Mai Nguyen, and Dominique Duhaut, 'Learning a set of interrelated tasks by using sequences of motor policies for a strategic intrinsically motivated learner', in *Proceedings of IEEE International Conference on Robotic Computing*, (2018).

[3] Timon Gehr, Matthew Mirman, Dana Drachsler-Cohen, Petar Tsankov, Swarat Chaudhuri, and Martin T. Vechev, 'AI2: safety and robustness certification of neural networks with abstract interpretation', in *IEEE Symposium on Security and Privacy*, (2018).

[4] Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum, 'Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation', in *NeurIPS*, volume 29, (2016).

[5] Siyuan Li, Lulu Zheng, Jianhao Wang, and Chongjie Zhang, 'Learning subgoal representations with slow dynamics', in *ICLR*, (2021).

[6] Alexandre Manoury, Sao Mai Nguyen, and Cédric Buche, 'Hierarchical affordance discovery using intrinsic motivation', in *HAI*. ACM, (2019).

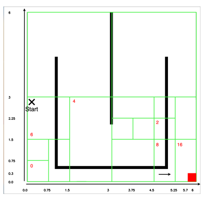
[7] Ofir Nachum, Shixiang Gu, Honglak Lee, and Sergey Levine, 'Data-efficient hierarchical reinforcement learning', in *NeurIPS 2018*, (2018).

[8] Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu, 'Feudal networks for hierarchical reinforcement learning', *CoRR*, **abs/1703.01161**, (2017).

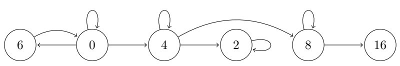
[9] Mehdi Zadem, Sergio Mover, and Sao Mai Nguyen, 'Goal space abstraction in hierarchical reinforcement learning via set-based reachability analysis', in *2023 IEEE International Conference on Development and Learning (ICDL)*, pp. 423–428, (Nov 2023).

[10] Mehdi Zadem, Sergio Mover, and Sao Mai Nguyen, 'Reconciling spatial and temporal abstractions for goal representation', in *The Twelfth International Conference on Learning Representations*, (2024).

[11] Tianren Zhang, Shangqi Guo, Tian Tan, Xiaolin Hu, and Feng Chen, 'Generating adjacency-constrained subgoals in hierarchical reinforcement learning', in *NeurIPS*, (2020).



(a) Subgoals decomposition



(b) Extracted graph representing the transitions between regions

Figure 4: Learned representation of GARA on the 2 Paths Maze and the planning inferred inferred as a graph between sugoal regions.