Noname manuscript No.
(will be inserted by the editor)

Robots Learn Increasingly Complex Tasks with Intrinsic Motivation and Automatic Curriculum Learning

Domain Knowledge by Emergence of Affordances, Hierarchical Reinforcement and Active Imitation Learning

Sao Mai Nguyen [1,2,5] · **Nicolas Duminy** [3,5] · **Alexandre Manoury** [2,5] · **Dominique Duhaut** [3,5] · **Cedric Buche** [4,5]

Received: date / Accepted: date

Abstract Multi-task learning by robots poses the challenge of the domain knowledge: complexity of tasks, complexity of the actions required, relationship between tasks for transfer learning. We demonstrate that this domain knowledge can be learned to address the challenges in life-long learning. Specifically, the hierarchy between tasks of various complexities is key to infer a curriculum from simple to composite tasks. We propose a framework for robots to learn sequences of actions of unbounded complexity in order to achieve multiple control tasks of various complexity. Our hierarchical reinforcement learning framework, named SGIM-SAHT, offers a new direction of research, and tries to unify partial implementations on robot arms and mobile robots. We outline our contributions to enable robots to map multiple control tasks to sequences of actions: representations of task dependencies, an intrinsically motivated exploration to learn task hierarchies, and active imitation learning. While learning the hierarchy of tasks, it infers its curriculum by deciding which tasks to explore first, how to transfer knowledge, and when, how and whom to imitate.

Keywords Intrinsic motivation · Continual learning · Curriculum learning · Transfer learning · Multi-task learning · Hierarchical reinforcement learning

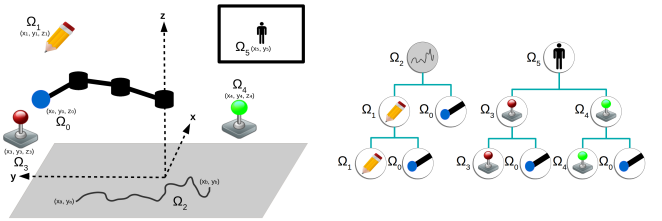


Fig. 1: Setup1: a robotic arm, can interact with the different objects in its environment (a pen and two joysticks). Both joysticks enable to control a video-game character. The pen can be used to draw. Left: the simulation environment. Right: the corresponding hierarchy of models

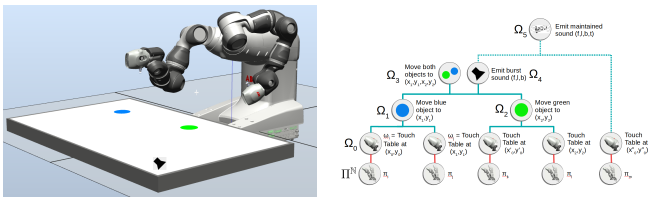


Fig. 2: Setup2: the robot arm can produce sounds by moving the blue and green objects

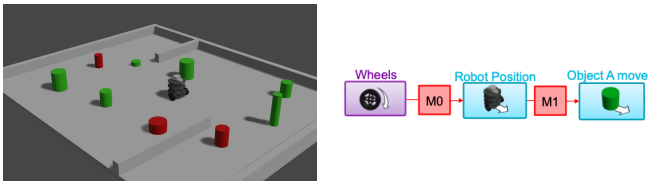


Fig. 3: Setup3: the mobile robot can avoid red obstacles, move green objects, push green objects with other objects. The objects are of random sizes.

1 Introduction

In the mainstream approaches based on classical artificial intelligence and machine learning, robotic engineering approaches have made several valuable application-specific impacts. Yet, the achievements are often subject to restrictions that involve domain knowledge, a bounded and specific environment, or a limited set of tasks of the same complexity.

To face the challenges of multi-task learning in a continual manner for embodied agents interacting with their stochastic environment and with humans, methods have taken inspiration in the living, and especially in how adults and infants learn as they develop, adapt and create new skills all along their lives. These methods fall into the field named *cognitive developmental robotics* [1, 6], within which we examine representations of actions and task relationships, their application to affordance learning and learning methods based on autonomous and socially guided exploration.

[1] Flowers team, U2IS, ENSTA Paris, Institut Polytechnique de Paris & Inria, France, [2] IMT Atlantique, Brest, France, [3] Université Bretagne Sud, Lorient, France, [4] ENIB, Brest, France, [5] Lab-STICC, UMR 6285, team RAMBO
E-mail: nguyensmai@gmail.com

Nguyen, S.M., Duminy, N., Manoury, A. et al. Robots Learn Increasingly Complex Tasks with Intrinsic Motivation and Automatic Curriculum Learning. Knstl Intell 35, 8190 (2021). https://doi.org/10.1007/s13218-021-00708-8

1.1 Learning Sequences of Motor Policies

In the case of multiple control tasks (ie. induced by actions) with various complexities and dimensionalities, the complexities of the actions required to complete them should be unbounded, without a priori knowledge. If we relate the complexity of actions to their dimensionality, actions of unbounded complexity should belong to spaces of unbounded dimensionality. For instance in setup Fig.1, if an action of dimensionality n is sufficient to draw a letter, a sequence of 2 such actions, i.e. an action of dimensionality $2n$ is sufficient to draw 2 letters. Generally, texts have variable lengths, thus there is no bound to the length of the sequence of actions. Likewise, an unbounded sequence of actions is needed to play tunes of any length in setup Fig.2; and to move all green objects side by side in setup Fig.3. Hence, here, we consider actions of unbounded complexity and suppose that they can be expressed as a sequence of action primitives. We consider *action primitives* and *sequences of actions*, also named in [38] respectively *micro actions* and *compound actions*. The agent thus needs to estimate the complexity of the task and deploy actions of corresponding complexity.

To tackle the curse of dimensionality which increases as sequences of actions grow longer, methods such as *DQN* [24], using a deep neural network architecture have successfully handled high dimensional continuous outcome and context spaces but still with discrete action spaces. Mnih et al. [25] proposed an asynchronous variant of the *Actor Critic* algorithm, relying both on deep neural networks and gradient policies. It successfully handles continuous action spaces, but still of predefined dimensionality. The *options* framework proposes a temporally abstract representation of actions [33], leading to sequences of actions for later reuse. Approaches combining *options* with TD networks enable compositional prediction [34]. Learning simple skills and planning sequences of actions instead of learning a sequence directly has been proposed as *Skill chaining* [19]. This forward chaining was successfully implemented to generate multi-step plans for robot affordance learning [36].

Following the ideas of a temporally abstract representation of actions and of multi-step planning, we propose a goal-directed representation of compound actions and a multi-step plan using inverse chaining of self-discovered subtasks.

1.2 Task Hierarchy for Curriculum Learning

This requirement of unbounded complexity of actions stems from the objective of learning unknown multiple tasks. Multi-task learning in biological agents is progressive and continual. Humans and other species develop and create new skills all along their lives as they adapt to their environment and to their own needs. In particular, infants learn skills of increasing level of difficulty as they grow up: mastering simple skills first and then learning more complex skills based on the previous simple ones. Indeed, in multi-task learning problems, some tasks can be compositions of simpler tasks, which we call *'complex tasks´* or *'composite tasks´*. The learning agent should be *starting small* before trying to learn more complex tasks, as phrased in [15]. Devising the order in which tasks should be learned has been coined *'curriculum learning'* in [5]: a learning agent needs to decide at each episode both which tasks it wants to learn to control (goal) and which actions to try (means). In our works, we thus take the hypothesis that tasks can be hierarchically related, some may be considered as subtasks of more complex tasks (*'hierarchically organised tasks´*). We conjecture that this hierarchy can help bootstrap the learning, by transfer of knowledge from simple to complex tasks.

Indeed, when given a task hierarchy, the robot can exploit this domain knowledge to reuse previously acquired skills to build more complex ones for tool use, as shown in [10]. Approaches for hierarchical multi task learning with neural networks have also been proposed, such as *Hierarchical DQN* [20], that uses intrinsic motivation to train a neural network in a fixed hierarchical manner.

To depend less on domain knowledge, the works presented in this article seek to learn the hierarchy between tasks, by exploring the different combinations between tasks. While learning the dependencies between tasks, we show in [14], that reusing the knowledge of simple tasks as subgoals for more complex tasks indeed greatly reduced the exploration, and in [22], that planning can be used in combination of emerging hierarchical models of tasks.

1.3 Emergence of Hierarchical Affordances

An application case of motor learning and task hierarchy is affordances learning. The concept of *affordance* has been first introduced by Gibson in [17] to characterise physical states in an action-oriented fashion, in terms of the possible interactions an agent may have with objects. Even without knowing an object specifically, seeing visual cues of a handle may suggest possible embodied interactions.

In affordances learning, many approaches have been developed [18]: for instance, the traversability affordance has been studied in different works [23]. Likewise, the grasp affordance is a recurrent topic and various approaches exist to learn it such as learning based on visual descriptors or raw image input [26, 21]. However such methods focus on one, or a fixed number of specific affordances, with no mechanism adapting it to new or more complex affordances.

We aim to continual learning of multiple affordances through the interaction with its environment. Thus, the robot builds sensory motor skills using a wide variety of actions. The robot can use actions of unbounded length and duration,

Nguyen, S.M., Duminy, N., Manoury, A. et al. Robots Learn Increasingly Complex Tasks with Intrinsic Motivation and Automatic Curriculum Learning. Knstl Intell 35, 8190 (2021). https://doi.org/10.1007/s13218-021-00708-8

in a continuous action space. Ugur and Piater [35] proposed an emergence of a hierarchical structure of affordances. However, affordances were defined as a list of discrete effects on objects, and the actions are manually coded. We would like to tackle continuous features of affordances and be able to learn compound actions in continuous action spaces. We extended their work with intrinsic motivation and planning.

1.4 Intrinsic Motivation as an Exploration Heuristic

To allow multi-task learning, developmental methods have transposed into algorithms the notion of intrinsic motivation, that has been outlined as a key mechanism for exploration [9]. These methods use a reward function that is not shaped to fit a specific task but is general to all tasks the robot will face. Tending towards life-long learning, this approach, also called artificial curiosity, may be seen as a particular case of reinforcement learning using an intrinsic reward function.

Methods based on *Q-Learning* and intrinsic motivation have been proposed in [37] for discrete environments where the reward depends on how much new information have been acquired. Other methods used intrinsic motivation to explore the action space of the robot, based on empirical measures of prediction progress, such as algorithms *IAC* [31] and its active learning version *RIAC* [2]. Then Baranes and Oudeyer [3] added goal-babbling to explore the outcome space to address higher dimensional action spaces with the *SAGG-RIAC* algorithm. More recently, intrinsic motivation and goal babbling have combined deep neural networks and replay mechanisms for automatic curriculum learning of multiple tasks of different complexities. IMGEP [16] – a formalisation of unsupervised multi-goal RL – GEP-GP [7] – mixing evolutionary methods and deep RL – and CURIOUS [8] – mixing parametrised reward function and automated curriculum learning – could select goals in a developmental manner from easy to difficult tasks.

1.5 Active Imitation Learning

Methods taking advantage of human demonstrations have shown to tackle more varied and large goal spaces. They were combined with intrinsic motivation such as in [29]. The bootstrapping effect is all the more efficient when the learning robot uses active learning based on intrinsic motivation to choose what to learn, and who, when and how to imitate. This choice on the source of information, has been called a *active imitation learning* and corresponds to the psychological description of infants' selectivity of social partners in [4]. The SGIM-ACTS algorithm proposed in [28] for multi-task learning has been applied to 3D object recognition [30] or mother tongue imitation by a vocal tract [27].

1.6 Summary and Position

Grounding our studies in cognitive developmental robotics, we aim for a robot capable **of discovering and learning multiple tasks as well as its curriculum by leveraging the relationships between tasks**. In this article, we show that domain knowledge about task relationship and complexity can be learned in order to adapt the complexity of the compound actions (sequences of actions) required. The task relationships between known or emerging tasks can be discovered with intrinsically motivated exploration or active imitation. We propose a common algorithmic architecture based on intrinsically motivated exploration to implement these mechanisms. In the following sections, we present a formalisation of the problem of hierarchical learning, two representations of task hierarchy and a unifying algorithmic architecture with 3 partial implementations [12, 14, 22]. We show how they address the following questions:

- how can knowledge from easier tasks be transferred to complex tasks while the robot learns tasks relationships?
- how can a robot discover new learnable tasks from which to transfer knowledge to more complex tasks?
- how can human demonstrations be beneficial to an active learning robot tackling multiple control tasks?

Compared to [11], we have made the formalisation more coherent and have updated the implementations and experimental setups taken from [14, 22], and position these works better in comparison with existing state-of-the-art.

2 Hierarchical Learning Framework

In this section, we propose a formalisation of the problem of learning compound actions to achieve hierarchically organised tasks, and propose an algorithmic architecture based on intrinsic motivation to learn the curriculum, common to the algorithms IM-PB, CHIME and SGIM-PB [12, 14, 22].

2.1 Formalisation

Let us consider a robot interacting with a non-rewarding environment by performing sequences of motions of unbounded length in order to induce changes in its surroundings.

Each of these motions is named a *primitive action*, described by a parametrised function with p parameters: $a \in \mathcal{A} \subset \mathbb{R}^p$. We call $\mathcal{A}$ the *primitive action space*. Our robot can perform sequences of primitive actions. Let a *compound action* be a sequence of any length $n \in \mathbb{N}$ primitive actions, and be described by $n * p$ parameters : $a = [a_1, \ldots, a_n] \in \mathcal{A}^n$. Thus the action space exploitable by the robot is a continuous space of infinite dimensionality $\mathcal{A}^{\mathbb{N}} \subset \mathbb{R}^{\mathbb{N}}$.

The actions performed by the robot have consequences on its environment, which we call outcomes $\omega \in \Omega$, where

Ω is a subspace of the state space S defining the control tasks to learn. Once the robot knows how to cause an outcome ω, we say the outcome is then *controllable*. The set of controllable outcomes is $\Omega_{cont} \subset \Omega$ and this set changes as the robot learns new tasks. For convenience, we define the *controllable space* $\mathcal{C} = \mathcal{A} \cup \Omega_{cont}$, regrouping both primitive actions $\mathcal{A}$ and observables that may be controlled, Ω_{cont}.

The robot learns tasks/models T each mapping controllable $c \in \mathcal{C}_T \subset \mathcal{C}$ and outcomes $\omega \in \Omega_T \subset \Omega$ within a given context $s \in S_T \subset S$. More formally, a task is a set of: a **forward model** $M_T : (S_T, \mathcal{C}_T) \rightarrow \Omega_T$ and an **inverse model** $L_T : (S_T, \Omega_T) \rightarrow \mathcal{C}_T$. The forward model is used to predict the observable consequence ω from a given context s of a controllable c, which is either a primitive action or a goal state that can be induced by a compound action. Conversely, the inverse model is used to estimate a controllable $\tilde{c}$ to be performed in a given context s to induce a goal observable state $\tilde{\omega}$: if $\tilde{c}$ is a primitive action the robot executes the action, otherwise it sets $\tilde{c}$ as a goal state and infers the necessary actions using other inverse models. We note that this definition of task T assumes that all $\omega \in \Omega_T$ can be realised independently from other observables, ie. the task dimensions of Ω_T do not interact with the rest of the state space in any way. In our works, we assume that the creation of the tasks (either predefined for IM-PB and SGIM-PB, or emerging for CHIME) ensures this coherence.

These models are trained on the data recorded by the robot along its exploration in its dataset $\mathcal{D}$. We define a strategy σ any process for exploration. For instance, we consider autonomous exploration and imitation learning strategies. Formally, we define it as a data collection heuristics, based on the current data, that outputs a set of triplets of initial state, action and outcome: $\sigma : \mathcal{D} \mapsto \{(s, a, \omega)\}$.

Let us also note $\mathcal{H}$ the hierarchy of the models used by our robot. $\mathcal{H}$ is formally defined as a directed graph where each node is a task T and its successors are the components of $\mathcal{C}_T$. As our robot learns this hierarchy, $\mathcal{H}$ varies along time. Its representation is detailed in section 3.

2.2 Algorithmic Architecture

We describe a generic algorithm Socially Guided Intrinsic Motivation for Sequence of Actions through Hierarchical Tasks (SGIM-SAHT) that learns and takes advantage of task hierarchy to solve increasingly complex tasks (Fig.4, Alg.1).

SGIM-SAHT learns by episodes in which a task T to work on, a goal outcome $\omega_g \in \Omega_T$ and a strategy σ have been selected to optimize progress and according to an interest map (see below). The selected strategy σ applied to the chosen goal outcome ω_g chooses a sequence of controllables $l_c = [c_1, \ldots, c_m]$ as a candidate to reach the ω_g (Alg.1, l.3).

SGIM-SAHT chooses an adequate task T_i , i.e. when the input space of L_{Ti} includes (s, ω_g). Then it applies L_{Ti} to find the action $a^i = L_{Ti}(c_i)$. This inference process may be recursive until the output is an action, using the hierarchy between tasks. SGIM-SAHT thus infers from l_c a compound action $a = [a_1, \ldots, a_n] \in \mathcal{A}^{\mathbb{N}}$, to be executed by the robot. The trajectory of the episode with the primitive actions and controllables sequence and goal and reached outcomes ω_g, ω_r are recorded in the memory (Alg.1, l.4).

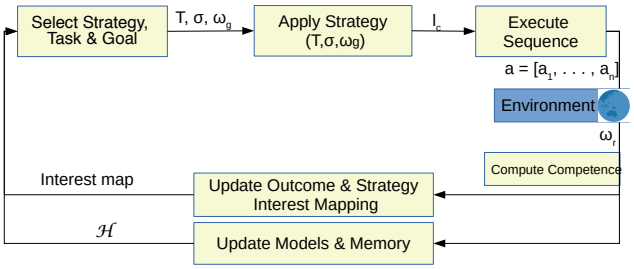


Fig. 4: The SGIM-SAHT algorithmic architecture

Algorithm 1 SGIM-SAHT

Input: the different strategies $\sigma_1, ..., \sigma_n$
Input: the initial model hierarchy $\mathcal{H}$
Initialization: partition of outcome spaces $R \leftarrow \bigsqcup_i \{\Omega_i\}$
Initialization: episodic memory *Memory* $\leftarrow \emptyset$
1: **loop**
2: $\sigma, T, \omega_g \leftarrow$ Select Strategy, Task & Goal Outcome$(R, \mathcal{H})$
3: $l_c \leftarrow$ Apply Strategy(σ, ω_g)
4: $\mathcal{D} \leftarrow (\omega_r, a, l_c) \leftarrow$ Execute Sequence(l_c)
5: $(comp(\omega_g), comp(\omega_r)) \leftarrow$ Compute Competence(ω_g, ω_r)
6: Update $M_T, L_T, \mathcal{H}$ with $(\mathcal{D}, comp(\omega_g), comp(\omega_r))$
7: $R_i \leftarrow$ Update Outcome and Strategy Interest Map$(R, \mathcal{D}, \omega_g)$
8: **end loop**

Then, it computes the learner's competence on the goal outcome. In the RL framework, this competence can be seen as the reward for the goal outcome. In our multi-task learning setting, we use as competence a reward function common to all goals based on the Euclidean distance between the goal outcome ω_g and the reached outcome ω_r (Alg.1, l.5). Variations of this metric have been implemented in [14, 22].

The memory and competence are used to update the models M_T and L_T, the set of tasks, and the hierarchy of models $\mathcal{H}$ (Alg.1, l.6). These mechanisms differ in our 3 algorithms. Besides, the competence is used to obtain an interest map that associates to each strategy and region of outcome space partition an interest measure to guide the exploration. The interest measure is computed as the progress or derivative of the competence of the enclosed outcomes (details in [28]).

When the number of outcomes added to a region R_i exceeds a fixed limit, the region is split into two regions with a clustering boundary that separates outcomes with low from those with high interest (details in [28]).

SGIM-SAHT has three implementations described in the sections 4 and 5. Their differences are described in table 1.

Algo.	Strategies σ	Tasks set	Comp. Act.
IM-PB	Outcomes, Procedures explo.	Static set	Procedures
CHIME	Action space, outcome space explo.	Dynamic set (emerging tasks)	Planning
SGIM-PB	Outcomes, Procedures explo.; active imitation	Static set	Procedures

Table 1: Differences between the 3 implementations of SGIM-SAHT

3 Task Hierarchy Representation

In this section, we describe two representations of task hierarchy, called Procedure and CHIME, to be used to learn the domain knowledge about the relationship between tasks.

3.1 Procedure Hierarchy Representation

The first is a goal-directed representation of action sequences in the form of sequences of subgoals or task decomposition, that enable transfer of knowledge between inter-related tasks. It is a temporally abstract representation of a succession of actions. More formally, **a procedure is defined as a succession of outcomes** $(\omega_1, \omega_2, ..., \omega_n)$. The succession is unbounded. The procedure space is $\Omega^{\mathbb{N}}$. A task decomposition into a procedure is an association of a goal outcome to a procedure. Thus, an outcome represents a task node in $\mathcal{H}$, while the task decomposition represents the directed edges and the procedure is the list of its successors. $\mathcal{H}$ is initialised as a densely connected graph, and the exploration prunes the connexions by testing which procedures or task decompositions respect the ground truth.

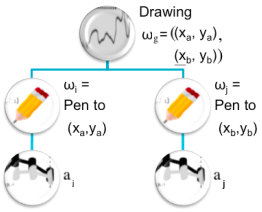


Fig. 5: Illustration of a procedure for setup fig.1. To make a drawing ω_g between points (x_a, y_a) and (x_b, y_b), a robot can recruit subtasks consisting in (ω_i) moving the pen to (x_a, y_a), then (ω_j) moving the pen to (x_b, y_b). These subtasks will be completed respectively with actions a_i and a_j. To complete this drawing, the learning agent can use the sequence of actions (a_i, a_j)

Executing a procedure $(\omega_1, \omega_1, ..., \omega_n)$ means building the action sequence a corresponding to the succession of actions $a_i, i \in [\![1, n]\!]$ (potentially action sequences as well) and execute it (where the a_i reach best the ω_i $\forall i \in [\![1, n]\!]$ respectively). Fig.5 illustrates this idea of task hierarchy.

The procedures representation is based on a static set of (ie. predefined) controllable and outcome spaces.

3.2 CHIME Hierarchy Representation

In comparison, the CHIME hierarchical representation is based on **a dynamic set of controllable and outcome spaces and the emergence of nested models**. It is built using simple models $M_T : (S_T, \mathcal{C}_i) \rightarrow \Omega_j$ which can rely on others: lower models map outcomes to actions while higher models map them to other outcomes that should be reached. For instance in the setup3 (Fig.3), the robot can move itself with the model $(M_0 : wheelCommand \in \mathcal{A} \rightarrow (x_0, y_0) \in \Omega_0)$ and it can learn that the position (x_1, y_1) of the object pushed depends on its own position $(M_1 : (x_0, y_0) \in \Omega_0 \rightarrow (x_1, y_1) \in \Omega_1)$. The combination of the nested models enable the robot to control (x_1, y_1) with its wheel commands (Fig.3 Right). In the hierarchy of models $\mathcal{H}$, the models M_T represent the directed graph from Ω_j to $\mathcal{C}_i$. At the beginning $H = \emptyset$, no model is present, and the robot chooses itself what model to create or modify: if $\mathcal{C}_i$ seems to be highly correlated to Ω_j it may create the model $M : \mathcal{C}_i \rightarrow \Omega_j$, which becomes the first nodes and edge of $\mathcal{H}$.

4 Learn Task Relationships by Autonomous Exploration

The idea of SGIM-SAHT is to use a hierarchical representation of the outcome and controllable spaces. This representation outlines the dependencies between tasks in order to reuse previous knowledge and use actions of adapted complexity. In this section, we examine how task hierarchy can be learned in the cases of a static and a dynamic set of tasks.

4.1 Learn Task Hierarchy from a Static Set of Tasks

Let us consider that the set of tasks is given. The robot needs to choose which are easier to learn first, which are more difficult, and which tasks can be reused as sub-goals for more difficult models. To learn the hierarchy $\mathcal{H}$ between tasks, we proposed in [13] an implementation, IM-PB (Intrinsically Motivated Procedure Babbling) based on procedures and the identification among all possible dependencies, of those that are valid.

IM-PB has 2 strategies : autonomous exploration of the outcome or the procedure spaces. Intrinsic motivation guides the exploration of the the task space and the procedure space (Alg.1, l.2) to find a curriculum from simple to complex tasks. IM-PB takes advantage of the dependencies between tasks : when executing a sequence (Alg.1, l.3), carrying out a procedure $(\omega_1, ...\omega_n)$ means carrying out the action primitive sequence π by executing sequentially each component action a_i, where a_i is an action that reaches ω_i $\forall i \in [\![1, n]\!]$.

The experiments on setups of Fig.1,2 in [12, 14] show that an intrinsically motivated learner is capable of learning sequences of motor actions. During the learning phase, results show that the robot explores mainly the task space for simple tasks, and it explores considerably more the procedure space for complex hierarchical tasks. Thus it implicitly

Nguyen, S.M., Duminy, N., Manoury, A. et al. Robots Learn Increasingly Complex Tasks with Intrinsic Motivation and Automatic Curriculum Learning. Knstl Intell 35, 8190 (2021). https://doi.org/10.1007/s13218-021-00708-8

understands that simple tasks do not need to be decomposed into subtasks and can be reached directly by action primitives. On the contrary for complex tasks, it is more advantageous to seek which subtasks to reuse. During the test phase, the robot uses the correct task decomposition. Furthermore, It can also adapt the length of its action sequence to the task to achieve: the results show that the length of the sequence of actions increases as the complexity of the task increases in terms of its hierarchy. **Combining these procedures with the learning of simple actions to complete simple tasks, it can build sequences of actions to achieve complex tasks**. We showed in [14] that the robot can take advantage of the procedures representation to improve its performance, especially on high-level tasks. It also adapts the complexity of its action sequence to the complexity of the task at hand.

Nevertheless, this adaptation is limited to the first two levels of task hierarchy, and the learner can not well adapt this complexity to a deeper hierarchy of tasks. To help the robot improve its understanding of task dependencies, we present in section 5 the benefits of active imitation learning.

4.2 Learn Task Hierarchy from a Dynamic Set of Tasks

In the previous section, the set of possible tasks (association of inputs and outputs) are given, and the robot needs to learn the relationship between them. In other terms, in $\mathcal{H}$, the nodes are pre-defined and the robot discovers the connections in the graph. In this section, we consider the case where the robot needs to learn both the connections and the nodes, ie. task hierarchy at the same time as task emergence.

4.2.1 Experimental Setup

It was applied to a wheeled robot with obstacles and movable objects of random size (Fig.3). It proposed an emergence of affordances : the algorithm is able to discover learnable models (eg. move a green object), and once the model is learned, its output can be used as input features of more complex models (eg. push an object with a green object), leading to hierarchical learning. In this paper, we describe an affordance as a task T.

4.2.2 CHIME Implementation

Let us consider that we do not have a set of sub-goals given, but have only given a high-dimensional set of inputs and observable outputs. At initialisation, the robot only has access to the control of its wheels, the positions and physical properties of objects. The set of tasks and controllable outcomes is empty. By exploration, it discovers changes in its environments (eg. green object moved) and how it can control them (eg. its own position), but also how these changes can induce more complex outcomes (eg. push an object with another).

The algorithm CHIME [22] implements SGIM-SAHT for a dynamic set of tasks and discovers new tasks by enactive exploration. At each episode, the physical properties such as colour, height, diameter of objects are generated randomly. The main characteristics of CHIME, detailed in [22], lie in its execution of sequence and its update of models and memory (Alg.1, l.4 and 6).

To execute a sequence of controllables l_c (Alg.1, l.4), for each element c_i :

- if c_i is a primitive action, it is directly executed
- if c_i is not a primitive action, $c_i \in \Omega_{cont}$. An affordance $T(S_T, \mathcal{C}_T, \Omega_T)$ is then selected (with $c_i \in \Omega_T$) and its inverse model is applied to obtain the controllable $b_i = L_T(c_i)$. If c_i is out of reach within a timestep, a planning phase is used to build a sequence of element of $\mathcal{C}_A$ in order to reach c_i. The same mechanism is applied recursively on it until having only primitive actions.

To update its models (Alg.1, l.6), at the end of each episode, subspaces of Ω for which outcomes ω has been observed are listed. Then the robot verifies if Ω matches a known affordance. To save computing time, we only verify for randomly selected subspaces of Ω. If it does not match, it creates a new affordance, ie. a forward and an inverse model. If it matches, but the new data contradicts with previous data (the competence for this task is reduced), it tries to update the model by adding context spaces.

4.2.3 Developmental Emergence of Affordances

The results on setup3 (Fig.3) show in [22] that CHIME discovers new affordances, and uses unbounded sequences of learned actions to complete all tasks. Even without predefinition of possible tasks, the agent is able to discover the inputs and outputs of models of learnable tasks, and how they are related to each other. We show in [22] that these tasks emerge and are learned in a developmental order from the lowest to the highest level of hierarchy. Planning based on these emergent tasks enables the robot to infer a sequence of actions to complete complex tasks.

The learning is based on active learning to collect data through new interactions with the environment, guided by the heuristics of intrinsic motivation. Once learned, these affordance control models are used to plan complex tasks with known or unknown objects, by using their physical properties to decide whether a learned affordance may be applied.

4.2.4 Automatic Curriculum Learning

Both IM-PB and CHIME rely on a temporally abstract representation of task relationships, whether all tasks have been predefined in advance or the robot has to discover new tasks. Discovering the task relationship enabled the robot to infer domain knowledge about the complexity of tasks and of

the actions needed, but also to build its curriculum, learning simple, then increasingly complex tasks.

5 Who, What, How to Imitate to Learn Task Relationships

Beyond mere autonomous intrinsically motivated exploration, we show that domain knowledge can also be learned through social guidance. In high dimensional and unbounded task spaces, the performance is improved even more when the robot can imitate actions and procedures from teacher demonstrations, when it actively chooses between self-supervised intrinsic motivation and imitation learning strategies.

For hierarchically organised tasks, we proposed in [14, 12] the implementation SGIM-PB (Socially Guided Intrinsic Motivation by Procedure Babbling) that merged IM-PB with SGIM-ACTS. In addition to the strategies σ of IM-PB, SGIM-PB has two strategies per teacher it can interact with : request a demonstration of actions or procedures. With the strategy *mimicry of an action*, SGIM-PB requests a demonstration of an action to reach ω_g. To apply the strategy (Alg.1, l.3), SGIM-PB adds noise to the demonstrated action parameters to explore locally the action parameters space, before executing the action (Alg.1, l.4).

With the strategy *mimicry of a procedure*, the learner requests a procedure for ω_g. SGIM-PB executes the demonstrated procedure $(\omega_{di}, \omega_{dj})$ by adding noise to the parameters, thus exploring locally the procedure space.

Using the competence measures and the interest map, SGIM-PB chooses when imitation learning is more beneficial than autonomous exploration, who among the different teachers are most expert in the field of knowledge it needs at the moment, and what kind of demonstrations is most beneficial. In terms of imitation learning, SGIM-PB self-determines who, what and when to imitate. Through setups Fig.1,2, we show in [14, 12] that demonstrations of procedures bootstrap the learning for tasks of the highest level of hierarchy. Moreover, demonstrations seem the most beneficial when they are demonstrations of policies for simple tasks, and when they are indications of procedures (i.e. subtasks) for complex tasks.

6 Conclusion

Through this article, we have presented three implementations of a common algorithmic framework for learning multiple control tasks through curriculum learning by discovering domain knowledge, such as the dependencies between tasks or task and action complexities, and exploiting this hierarchy to transfer knowledge from the easy tasks to the compositional tasks. Table 2 summarises the properties of IM-PB, SGIM-PB and CHIME in learning to perform complex tasks with compound actions, in contrast to the state of the art. While IM-PB and SGIM-PB rely on a static set of controllable and outcome features and explore the dependencies between tasks to learn sequences of actions, CHIME builds dynamically its set of tasks from emergent control models that are then used to plan sequences of actions. Whereas IM-PB and CHIME rely only on autonomous exploration using intrinsic motivation, SGIM-PB can request different kinds of demonstrations depending on the complexity of the target task to the appropriate teacher. All three rely on a temporally abstract representation of compound actions using task hierarchy. They efficiently manage to learn them through the discovery of relationships between tasks to enable transfer of knowledge.

Algorithm	Reward	Cont. goal	Multi-task	Hierarchy	Actions	Imitation
Qlearning [32]	Ext.				Discrete	
Qlearning & curiosity [37]	Int.				Primitive	
Option TDNet[34]	Ext.				Option	
Skill-chaining [19]	Ext.			Yes	Option	
DQN [24]	Ext.				Discrete	
h-DQN [20]	Int.		Yes	Yes	Primitive	
Asynch. Actor Critic [25]	Ext. self-eval				Primitive	
GEP-PG [7]	Int.	Yes	Yes			
CURIOUS [8]	Int.	Yes	Yes		Primitive	
IAC[31], RIAC[2]	Int.	Yes	Yes		Primitive	
SAGG-RIAC [3]	Int.	Yes	Yes		Primitive	
IMGEP [16]	Int.	Yes	Yes		Primitive	
SGIM-ACTS [28]	Int.	Yes	Yes		Primitive	Yes
IM-PB [13, 14]	Int.	Yes	Yes	Yes	Proced.	
SGIM-PB [12, 14]	Int.	Yes	Yes	Yes	Proced.	Yes
CHIME [22]	Int.	Yes	Yes	Yes	Planning	

Table 2: Comparison between the algorithms on : intrinsic vs extrinsic reward, the goal space is continuous (parametrised), single task vs multi-task problem, hierarchical learning, the action representation and whether imitation learning is used.

We have proposed a framework unifying their common aspects: a temporally abstract representation of the relationship between tasks, learning the hierarchy of tasks with intrinsically and socially guided exploration. This summary opens a theoretical blueprint of a novel framework for robots automatic curriculum learning of uncovering domain knowledge to learn compositional tasks. We showed this can be tackled by combining hierarchical learning, planning and exploration based on intrinsic motivation and active imitation learning.

In future works, we shall develop an implementation of this unified framework using all the described features: learning primitive actions and then planning sequences of them, then once learned, optimising directly these sequences owing to the procedure framework. The emergent subtasks will reduce the dependency on domain knowledge, whereas learning a representation of a compound action will result in better optimised policies and reduce the planning complexity.

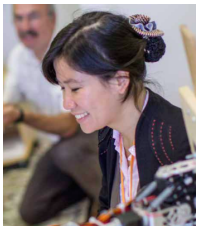
Nguyen, S.M., Duminy, N., Manoury, A. et al. Robots Learn Increasingly Complex Tasks with Intrinsic Motivation and Automatic Curriculum Learning. Knstl Intell 35, 8190 (2021). https://doi.org/10.1007/s13218-021-00708-8

Acknowledgements This work is partially supported by the European Regional Development Fund (ERDF) via the VITAAL CPER, by Institut Mines Telecom (IMT) and by the French Ministry of Research.

References

1. Asada M, MacDorman KF, Ishiguro H, Kuniyoshi Y (2001) Cognitive developmental robotics as a new paradigm for the design of humanoid robots. Robotics and Autonomous Systems 37(2-3):185–193
2. Baranes A, Oudeyer Py (2009) R-IAC: Robust intrinsically motivated exploration and active learning. IEEE Transactions on Autonomous Mental Development 1(3):155–169
3. Baranes A, Oudeyer PY (2013) Active learning of inverse models with intrinsically motivated goal exploration in robots. Robotics and Autonomous Systems 61(1):49–73
4. Begus K, Southgate V (2018) Active Learning from Infancy to Childhood, Springer International Publishing, Cham, chap Curious Learners: How Infants' Motivation to Learn Shapes and Is Shaped by Infants' Interactions with the Social World, pp 13–37
5. Bengio Y, Louradour J, Collobert R, Weston J (2009) Curriculum learning. In: International Conference on Machine Learning, ACM, New York, NY, USA, ICML '09, pp 41–48
6. Cangelosi A, Schlesinger M (2015) Developmental robotics: From babies to robots. MIT press
7. Colas C, Sigaud O, Oudeyer PY (2018) GEP-PG: Decoupling Exploration and Exploitation in Deep Reinforcement Learning Algorithms. In: ICML, Stockholm, Sweden
8. Colas C, Fournier P, Chetouani M, Sigaud O, Oudeyer PY (2019) CURIOUS: Intrinsically motivated modular multi-goal reinforcement learning. In: International Conference on Machine Learning, PMLR, Long Beach, California, USA, vol 97, pp 1331–1340
9. Deci E, Ryan RM (1985) Intrinsic Motivation and self-determination in human behavior. Plenum Press, New York
10. Duminy N, Nguyen SM, Duhaut D (2016) Strategic and interactive learning of a hierarchical set of tasks by the Poppy humanoid robot. In: ICDL-EPIROB
11. Duminy N, Manoury A, Nguyen SM, Buche C, Duhaut D (2018) Learning sequences of policies by using an intrinsically motivated learner and a task hierarchy. In: Workshop on Continual Unsupervised Sensorimotor Learning, ICDL-EpiRob, Tokyo, Japan
12. Duminy N, Nguyen SM, Duhaut D (2018) Effects of social guidance on a robot learning sequences of policies in hierarchical learning. In: IEEE (ed) International Conference on Systems Man and Cybernetics
13. Duminy N, Nguyen SM, Duhaut D (2018) Learning a set of interrelated tasks by using sequences of motor policies for a strategic intrinsically motivated learner. In: IEEE International on Robotic Computing, pp 288–291
14. Duminy N, Nguyen SM, Duhaut D (2019) Learning a set of interrelated tasks by using a succession of motor policies for a socially guided intrinsically motivated learner. Frontiers in Neurorobotics 12:87
15. Elman J (1993) Learning and development in neural networks: The importance of starting small. Cognition 48:71–99
16. Forestier S, Mollard Y, Oudeyer P (2017) Intrinsically motivated goal exploration processes with automatic curriculum learning. CoRR abs/1708.02190
17. Gibson J (1977) The theory of affordances. In: Perceiving, Acting, and Knowing, Robert Shaw and John Bransford
18. Jamone L, Ugur E, Cangelosi A, Fadiga L, Bernardino A, Piater J, Santos-Victor J (2016) Affordances in psychology, neuroscience, and robotics: A survey. IEEE Transactions on Cognitive and Developmental Systems 10(1):4–25
19. Konidaris G, Barto AG (2009) Skill Discovery in Continuous Reinforcement Learning Domains using Skill Chaining. Advances in Neural Information Processing Systems pp 1015–1023
20. Kulkarni TD, Narasimhan K, Saeedi A, Tenenbaum J (2016) Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In: Advances in neural information processing systems, pp 3675–3683
21. Levine S, Pastor P, Krizhevsky A, Ibarz J, Quillen D (2018) Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. The International Journal of Robotics Research 37(4-5):421–436
22. Manoury A, Nguyen SM, Buche C (2019) Hierarchical affordance discovery using intrinsic motivation. In: Human Agent Interaction
23. Mitriakov A, Papadakis P, Nguyen SM, Garlatti S (2020) Staircase traversal via reinforcement learning for active reconfiguration of assistive robots. In: International Conference on Fuzzy Systems (FUZZ-IEEE), pp 1–8
24. Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, Petersen S, Beattie C, Sadik A, Antonoglou I, King H, Kumaran D, Wierstra D, Legg S, Hassabis D (2015) Human-level control through deep reinforcement learning. Nature 518(7540):529–533
25. Mnih V, Badia AP, Mirza M, Graves A, Lillicrap TP, Harley T, Silver D, Kavukcuoglu K (2016) Asynchronous methods for deep reinforcement learning. CoRR abs/1602.01783
26. Montesano L, Lopes M (2009) Learning grasping affordances from local visual descriptors. In: 2009 IEEE 8th International Conference on Development and Learning, pp 1–6
27. Moulin-Frier C, Nguyen SM, Oudeyer PY (2014) Self-organization of early vocal development in infants and machines: The role of intrinsic motivation. Frontiers in Psychology 4(1006)
28. Nguyen SM, Oudeyer PY (2012) Active choice of teachers, learning strategies and goals for a socially guided intrinsic motivation learner. Paladyn Journal of Behavioural Robotics 3(3):136–146
29. Nguyen SM, Oudeyer PY (2014) Socially guided intrinsic motivation for robot learning of motor skills. Autonomous Robots 36(3):273–294
30. Nguyen SM, Ivaldi S, Lyubova N, Droniou A, Gerardeaux-Viret D, Filliat D, Padois V, Sigaud O, Oudeyer PY (2013) Learning to recognize objects through curiosity-driven manipulation with the icub humanoid robot. In: IEEE International Conference on Development and Learning - Epirob
31. Oudeyer PY, Kaplan F, Hafner V (2007) Intrinsic motivation systems for autonomous mental development. IEEE Transactions on Evolutionary Computation 11(2):265–286
32. Sutton RS, Barto AG (1998) Reinforcement Learning: an introduction. MIT Press
33. Sutton RS, Precup D, Singh S (1999) Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. Artificial Intelligence 112:181 – 211
34. Sutton RS, Rafols EJ, Koop A (2006) Temporal abstraction in temporal-difference networks. In: Weiss Y, Schölkopf B, Platt J (eds) Advances in Neural Information Processing Systems, MIT Press, vol 18, pp 1313–1320
35. Ugur E, Piater J (2016) Emergent structuring of interdependent affordance learning tasks using intrinsic motivation and empirical feature selection. IEEE Transactions on Cognitive and Developmental Systems
36. Ugur E, Piater J, Sahin E, Oztop E (2009) Affordance learning from range data for multi-step planning. In: International Conference on Epigenetic Robotics
37. Vigorito C, Barto A (2010) Intrinsically Motivated Hierarchical Skill Learning in Structured Environments. IEEE Transactions on Autonomous Mental Development 2(2):132–143
38. Zech P, Renaudo E, Haller S, Zhang X, Piater J (2019) Action representations in robotics: A taxonomy and systematic classification. International Journal of Robotics Research 38(5):518–562

Nguyen, S.M., Duminy, N., Manoury, A. et al. Robots Learn Increasingly Complex Tasks with Intrinsic Motivation and Automatic Curriculum Learning. Knstl Intell 35, 8190 (2021). https://doi.org/10.1007/s13218-021-00708-8



Sao Mai Nguyen specialises in robotic learning, especially cognitive developmental learning, reinforcement learning, imitation learning, curriculum learning for robots and human activity recognition. She received her PhD from Inria, an Engineer degree from Ecole Polytechnique and a master's degree from Osaka University, Japan. She has enabled a robot to coach physical rehabilitation in the projects RoKInter and the experiment KERAAL she coordinated, funded by the European Union FP-7 program. She is currently associate editor of the journal IEEE TCDS and co-chair of the Task force "Action and Perception" du IEEE Technical Committee on Cognitive and Developmental Systems.

Nicolas Duminy holds a master in engineering from IMT Atlantique in France and has obtained in 2018 his PhD in computer science from Universit Bretagne Sud in France. His research focused on developmental robotics, and more particularly on the strategic autonomous learning of action sequences and task hierarchies. Today, he is working as an engineer and entrepreneur to develop more immersive virtual reality experiences.

Alexandre Manoury hold a master in engineering from IMT Atlantique in France. His research focused on developmental robotics, and more particularly on the strategic autonomous planning of action sequences and task hierarchies.

Dominique Duhaut received his PhD degree in computer science from the University Paris 6 in 1982 on The study of complexity of recursive function under the definition of Trahtenbrot. He became assistant professor in 1984 in University Paris 6. In 1987, he moved to the Laboratoire de robotique de Paris where he worked on the cooperation of robots in a flexible cell. In 1996, he was participating to the definition of the RoboCup movement that he organised in Paris in 1998. At that time his research was focused on multi-agents programming in robotics. He moved in university of Bretagne Sud in 2000 where he became professor. His research interests are actually on self reconfigurable systems, teams of robot programming and social aspect of robots. He is also interested in science promotion for young people. He his organising competitions for schools since several years and is participating in the RoboFesta movement.

Cedric Buche is a professor at Ecole Nationale d'Ingnieurs de Brest in France. His research focuses on artificial intelligence and human-robot interactions. He is mainly interested in interactive machine learning approaches..