GRAIL: A Goal-Discovering Robotic Architecture for Intrinsically-Motivated Learning

Vieri Giuliano Santucci, Gianluca Baldassarre, and Marco Mirolli

***Abstract*—In this paper, we present goal-discovering robotic architecture for intrisically-motivated learning (GRAIL), a four-level architecture that is able to autonomously: 1) discover changes in the environment; 2) form representations of the goals corresponding to those changes; 3) select the goal to pursue on the basis of intrinsic motivations (IMs); 4) select suitable computational resources to achieve the selected goal; 5) monitor the achievement of the selected goal; and 6) self-generate a learning signal when the selected goal is successfully achieved. Building on previous research, GRAIL exploits the power of goals and competence-based IMs to autonomously explore the world and learn different skills that allow the robot to modify the environment. To highlight the features of GRAIL, we implement it in a simulated iCub robot and test the system in four different experimental scenarios where the agent has to perform reaching tasks within a 3-D environment.**

***Index Terms*—Autonomous robotics, developmental robotics, goal formation, hierarchical architecture, intrinsic motivations (IMs), reinforcement learning.**

I. Introduction

ARTIFICIAL agents are continuously improving. More and more sophisticated robots and powerful algorithms able to solve increasingly complex tasks are being developed every year. However, the autonomy and versatility of current artificial systems are still extremely limited in comparison to those of biological agents (humans in particular). While this is not a problem when artificial agents are used to solve predefined behaviors (e.g., for industrial robots), the lack of autonomy in present robots prevents them from properly interacting with real environments where they have to face problems that are unpredictable at design-time and where it is not clear which skills will be suitable to solve them [92].

Future robots have to be versatile, capable of managing different scenarios and form ample repertoires of skills that can be reused to solve different tasks [1], [46]. Beside being an interesting challenge per se, developing truly autonomous robots can enhance the effectiveness of robot exploitation. If we think at artificial agents exploring new planets or the deep abysses of the oceans, it is clear that these robots have to be designed so that they do not need to rely totally on designers' knowledge since most of the situations they encounter are completely unforeseeable. And even in more familiar settings, such as houses or other human environments, the capabilities to adapt to novel situations and to autonomously acquire new knowledge and skills are fundamental features for future robots.

For these reasons, we need to provide artificial agents with the capacity to autonomously discover new actions and to self-determine which goals to focus on so to learn the proper skills to accomplish them. Since biological agents are versatile and naturally-adaptive, looking at the characteristics that allow them to improve their knowledge and their competence could provide useful hints to develop intelligent artificial systems with similar capabilities [23].

A. Intrinsic Motivations

Humans and other mammals (e.g., rats and monkeys) explore the environment and learn new skills not only following the drives provided by reward-related stimuli (such as food, sex, etc.) but also in the absence of direct biological pressure [5]. In particular, novel or unexpected neutral stimuli are able to modify the behavior of biological agents and guide the acquisition of new skills [37]. The mechanisms related to these processes have been studied since the 1950s under the name of intrinsic motivations (IMs), first in animal psychology [21], [29], [54], [93] and then in human psychology [17], [18], [24], [70]. In the last decades, new research in the field of neuroscience [26], [64], [65], [95] has highlighted some of the neural mechanisms that seem to be involved in IM processes. On the basis of these data some bio-inspired/bio-constrained models have been developed [6], [19], [27], [34], [53], linking experimental evidence with computational mechanisms so to better understand the neural substrate of IMs. Other neuroscientific evidence [20], [81] reveal the importance of information-seeking behavior and mechanisms without directly referring to IMs. These phenomena have been also investigated with computational models [16], [35].

The insights provided by IMs have suggested to machine learning and autonomous robotics new strategies to implement autonomous agents [8]. In particular, IM learning signals

Manuscript received September 18, 2015; revised December 23, 2015; accepted February 8, 2016. Date of publication May 18, 2016; date of current version September 7, 2016. This work was supported in part by the European Commission under the 7th Framework Programme (FP7/2007- 2013), under the ICT Challenge 2 Cognitive Systems and Robotics, and in part by the Project IM-CLeVeR—Intrinsically Motivated Cumulative Learning Versatile Robots under Grant ICT-IP- 231722.

V. G. Santucci is with the Istituto di Scienze e Tecnologie della Cognizione, Consiglio Nazionale delle Ricerce, Rome 00185, Italy, and also with the School of Computing and Mathematics, University of Plymouth, Plymouth PL4 8AA, U.K. (e-mail: vieri.santucci@istc.cnr.it).

G. Baldassarre and M. Mirolli are with the Istituto di Scienze e Tecnologie della Cognizione, Consiglio Nazionale delle Ricerce, Rome 00185, Italy.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCDS.2016.2538961

provide a useful tool to drive the autonomous learning and selection of different skills without any assigned reward or task. Most of the IM computational models have been developed within the reinforcement learning framework [84]. Following the seminal machine learning works of [78] and [79] and the work of [12] most of these models implement IMs as intrinsic reinforcements based on some form of "information compression progress" [80], in particular the prediction error (PE), or the improvement in the PE (PEI), of a predictor that tries to anticipate the effects of the actions of the artificial system in the environment [10], [31], [43], [57], [60]. Other works have focussed on different IM mechanisms and functions: some focused on the autonomous acquisition of skills [12], [30], [49] and the autonomous selection of which skill to train [51], [74], [77], others on different aspects such as vision [45], speech development [55], or emotions [58].

Recently some theoretical works have focused on the different typologies of IMs [14], [52], [61] and the way to implement them [71]. In particular, this literature distinguishes between signals based on what the system perceives and knows, called knowledge-based IMs (KB-IMs), and signals based and what the system can do, called competence-based IMs (CB-IMs). Recent experimental evidence with robots [74] suggests that CB-IMs are more suitable than KB-IMs to drive an artificial system in the selection and acquisition of a repertoire of skills.

B. Importance of Goals

The use of CB-IM signals is strictly connected to the concept of goal, i.e., a particular state that the system is willing to achieve. Since competence is always competence in doing something, CB-IMs are related to the ability of the system in achieving a goal and are typically implemented as the PEI (or PE) of a predictor that is trying to anticipate the achievement of the desired state. Following this idea, in previous works we discussed [71] and showed [74] the importance of goals and CB-IMs in driving an artificial system in the autonomous acquisition of skills. IMs can play the role of guiding the selection of the task on which the agent is focussing its learning: as long as the system is improving its ability the IM signal will motivate the pursuit of the related goal, while when the agent has acquired the connected skill (or has learned that such skill can not be acquired) the IM signal will fade away and the system will be able to explore the environment and focus on different goals/skills.

The autonomous selection of goals and the autonomous learning of actions is a necessary step toward more versatile and adaptive agents. However, for a robot to be truly autonomous it is necessary to be able not only to select its own goals but also to discover them without designer intervention. In the "goal babbling" framework [67] the process of goal formation is tackled by focussing on the boost that goals can provide to learning based on the advantages of searching solutions within the task space rather than in the larger joint space [11], [68]. In these works goals are typically defined as every possible position of the effectors of the robot (e.g., of the terminal point of a robotic arm) and so are strictly connected to the body of the artificial agent.

Differently from goal babbling, in our previous works [71], [72], [74] and here, we consider goals that are not related to the position of the robot body in space but to the relation between the robot and the environment. Animals spend a great amount of time learning skills that modify the external world. We postulate that the reason of this is that what really counts for an agent is to acquire actions that allow it to have a strong impact on the world: knowing which are the effects of ones own actions on the environment is an important knowledge that can significantly improve the versatility and adaptation of biological organisms [91]. Moreover, empirical research (see [62], [65], [87]) has typically demonstrated that IMs are closely related to the unexpected modifications of the environment and the causes that generate them.

C. Overview

The present work focuses on building an artificial system that is autonomously able to discover changes in the environment and use them to drive the learning of new skills. To develop such complex functions, it is necessary that the system has an architecture that guarantees not only the selection of goals and the learning of the related skills but also the discovery and the formation of new goals. In a recent work [72], we implemented a three-level hierarchical architecture to control the two redundant arms of the simulated iCub robot [50] tested in a reaching task within a 3-D environment. That architecture allowed the system to both select the goals to pursue (through a CB-IM signal) and learn which was the best arm to use to achieve the different goals.

However, that system still lacked the capacity of autonomously discovering and forming representations of the different goals. Here we present goal-discovering robotic architecture for intrisically-motivated learning (GRAIL) (Section II-B)—a four-level architecture which allows the robot to: 1) autonomously form its own goals on the basis of the perceived changes in the environment; 2) store events as goals in internal representations; 3) autonomously select its own goals; 4) autonomously decide which computational resources to train to achieve each goal; 5) use the representations of the events to autonomously recognize the achievement of the selected goal; and 6) self-generate goal-matching (GM) signals that are used by the system both to motivate its goal-selection and to train the different levels of the architecture. To the best of our knowledge, points 1) and 2) represent novel mechanisms for the self-generation of goals. Mechanism 3) has been investigated also in other works, but with the differences underlined in Section I-B and further discussed in Section IV. The use of the processes 4)–6) constitutes a novelty in the field of autonomous open-ended learning. Finally, also the integration of all these mechanisms and processes represents a novel achievement in the field of autonomous robotics.

The overall objective of GRAIL is to allow the robot to autonomously discover different changes in the environment,

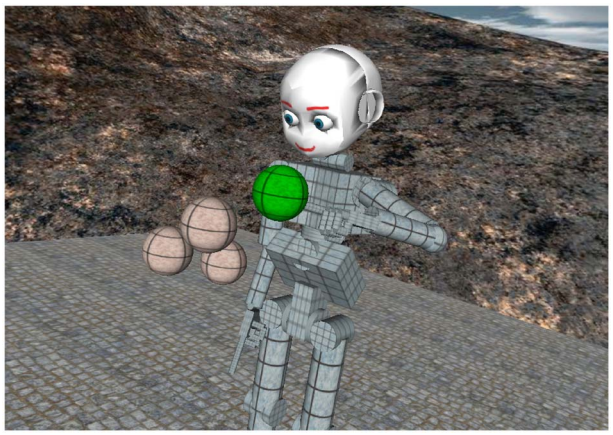


Fig. 1. Simulated iCub implemented with the FARSA simulator. The task consists in touching the different spheres positioned in front of the robot. When touched, the spheres change their color to green.

represent them as goals, and use these goals to guide the acquisition of the skills necessary to accomplish them. We implemented GRAIL in a simulated iCub and test its performance in four different experimental scenarios where we: 1) compare GRAIL with the previous architectures (Section III-B); 2) show the ability of GRAIL to autonomously discover new goals and adapt in possibly unknown scenarios with unexpected or changing goals (Section III-C); 3) test the system ability to ignore events that happen independently of robot activity (Section III-D); and 4) check the ability of GRAIL (together with IMs) to cope with stochastic environments (Section III-E).

II. Setup

In order to ease reading, the current section describes only the most relevant features of GRAIL and the experimental setup, whereas the technical details needed to fully implement the system and replicate the simulations can be found in the Appendix.

A. Robot and the Experimental Setup

To test the system we use a simulated iCub robot implemented with a 3-D physical engine simulator called FARSA (see [47] and http://laral.istc.cnr.it/farsa, Fig. 1). In the experiments presented in this paper, we use the two arms of the robot each with 4 degrees-of-freedom (DOFs) (the joints of the wrist and those of the fingers are kept fixed) in kinematic modality where collisions are not taken into account. The fingers of the two hands are closed, with the exception of the two forefingers that are kept straight and their tips used to establish when the robot touches an object in the environment.

Visual input is provided to the system by the right camera of the robot [Fig. 2(a)]: the eye is kept in a fixed position so that it can see all the targets. The task consists in reaching different fixed spherical objects with the fingertips of the forefingers. The objects are anchored to the world and have a radius of 4 cm: when a sphere is touched it becomes green [Figs. 1 and 2(b)].

B. Architecture and Methods

GRAIL is the last achievement of a series of increasingly complex architectures that we have been developing for autonomous open-ended learning. In [73], we implemented a two-level architecture composed of a "selector" that determines the goal to achieve, and a control layer with different components (the "experts") [7] that learn and store the skills associated to the goals. We referred to this first architecture as a coupled system (CS) since its selector presents a rigid coupling between goals and experts: each goal unit is associated to an expert, thus the selection of a goal automatically determines with which expert the robot pursues it. To improve CS, in [72] we developed a new three-level architecture which, decoupling the selection of the goals from the selection of the experts, is able to autonomously select both its goals and the computational resources to achieve them. We called this new architecture a decoupled system (DS) to underline the difference from the previous version, and we showed its ability to outperform CS in a reaching task where it was not clear which was the best arm (and hence expert) to use to achieve the different targets.

The DS (as well as the CS) needs to know in advance which are the possible goals to be achieved. This is a strong limitation if one wants a system that is able to autonomously interact with complex situations in real environments where it is not possible to determine at design time not only which skills will be useful for the robot but also which are the possible events that the system can produce. GRAIL does not only select which goal to achieve at each trial and the expert to use to pursue it, but it is also able to discover new goals, form representations of the events associated to the goals, and autonomously check if a goal is achieved [4]. The architecture (Fig. 3) is composed of four levels: 1) the goal-formation mechanism; 2) the goal-selector; 3) the expert-selector; and 4) the experts.

1) Goal-Formation Mechanism: The goal-formation mechanism is the main innovation of GRAIL with respect to our previous systems [72], [73] and existing open-ended learning architectures. As mentioned in Section I-B, we believe that to improve the autonomy of robots we need to develop an architecture that guides the system in learning new skills on the basis of the effects that the actions of the robot have on the external world. For this reason, we endow GRAIL with a novel mechanism that is able to autonomously recognize the changes in the environment and store them as possible goals. The robot is so able to discover new effects and autonomously select them through the goal-selector component (Section II-B2). Moreover, the storage of the discovered events allows the robot to autonomously check if it has achieved the goals that it was pursuing: confronting the "representation" (see below in this section) of the selected goal with the actual event that happens in the environment, the goal-formation mechanism is able to produce a GM signal that on the one hand indicates if the desired state (goal) has been achieved, and on the other hand represents the learning signal for the entire architecture (see Section II-C for a detailed description of the GM signal). The importance of the goal-formation and matching mechanisms, and the ability of the rest of the architecture to use them to learn and suitably select the skills related to the formed goals,

(a) (b) (c)

Fig. 2. Visual input is provided by the fixed right camera of the simulated robot. (a) Robot camera image of the environment in experiment 1. (b) Image after a change in the environment determined by the event of one sphere lighting up. (c) Binary image obtained by subtracting images (a) and (b). Since the image of the event is resized, the final input can be slightly different from the actual change seen in (b).

is that they can work with changes of the environment that are not envisaged at design time but are autonomously discovered by the robot with the exploration of the possible effects of its action on the environment.

The goal-formation mechanism receives a visual input related to the changes in the environment and allows the system to form representations of those events. This level is composed of two elements: 1) a winner-takes-all (WTA) competitive network [69] whose output, called implicit representation vector (IR-V), forms an abstract representation of the events and 2) a map, called the explicit representation map (ER-M), that stores the actual representations of the events. The events are identified as follows: at every time step, the image of the previous time step is subtracted from the current image (for simplicity the arms are ignored) so that when there is no change the resulting image is black (all zeros), while when a visual change has happened, the pixels corresponding to the change become white (ones). The resultant binary image becomes the input to the goal-formation mechanism: Fig. 2(c) is an example of the input provided by an event, determined by the difference between Fig. 2(a) and (b). The visual input determines a double effect: on the one hand it activates the IR-V (via all-to-all connections) determining through an Hebbian-like learning rule the association between different events and different output units (10 in these simulations), and on the other hand it activates (through one-to-one connections) the ER-M determining an activation that is topologically identical to the visual input. These two activations are used to modify through Hebbian learning the weights projecting from each unit of the IR-V to each corresponding unit of the ER-M. Gradually, the connections linking IR-V units, representing an event, to the ER-M units become able to generate an activation within it that is identical to the actual event representation in the visual input map. In this sense, we consider the pattern stored in those connections as a representation of the event that the system is able to reactivate when a goal is selected (Section II-B2), even when the event is not perceived.

2) Goal-Selector: The goal-selector is composed of N units (10 in the simulations presented in this paper) that project with fixed one-to-one connections to the IR-V. At the beginning of every trial, the goal-selector determines through a softmax selection rule [84] a winner unit. When a unit is selected it directly activates the corresponding IR-V unit. When the experiment starts the units are not associated to any goal while over time the system discovers new events and associates them to different units (Section II-B1). At the beginning of each trial the activation of each unit of the goal-selector is determined by an exponential moving average (EMA) of the intrinsic reinforcement (the CB-IM signal) for obtaining the goal that the goal-formation mechanism has associated to that unit. Since the CB-IM signal (Section II-D) is a measure of how much the system is improving its competence in achieving a certain goal, the system will select with a higher probability those goals whose skills have a high learning rate with respect to those that are not improving or are improving less. Note that we are using the IM signal for motivation (i.e., as a drive to select and execute a goal/behavior) rather than for learning (i.e., to update connection weights) as in most models using IMs.

3) Expert-Selector: The selector of the experts is composed of $M \times N$ units (in the experiment presented in this paper $M = 8$). Each goal unit is connected to M units, each of which corresponds to one expert. Half of the experts control the right arm and half controlling the left arm. Every goal can be pursued through any expert. At the beginning of every trial, the winning unit of the goal-selector determines which set of units of the expert-selector is active. The activation of the selected set of units is determined by an EMA of the GM signal (see Section II-C). The activation of these units is used to select the expert to use in the current trial with a softmax function.

4) Experts: The experts (8 in these simulations) are implemented as actor-critic networks [15] modified to work with continuous states and actions spaces [25], [76], although we did not modify the core formulas of the reinforcement learning algorithm as done in [25]. The input to each expert are the angles of the four actuated joints of the related arm (three joints for the shoulder, one for the elbow). The actor of each expert has four output units whose activation, with the addition of noise, is used to determine the motor command sent to each joint of the active arm (the joints are controlled in velocity). The expert selected to control the robot in the current trial is trained through a TD reinforcement learning algorithm [84].

C. Goal-Matching Signal

GRAIL is able not only to recover the explicit representation of the goal it is trying to achieve, but also to autonomously

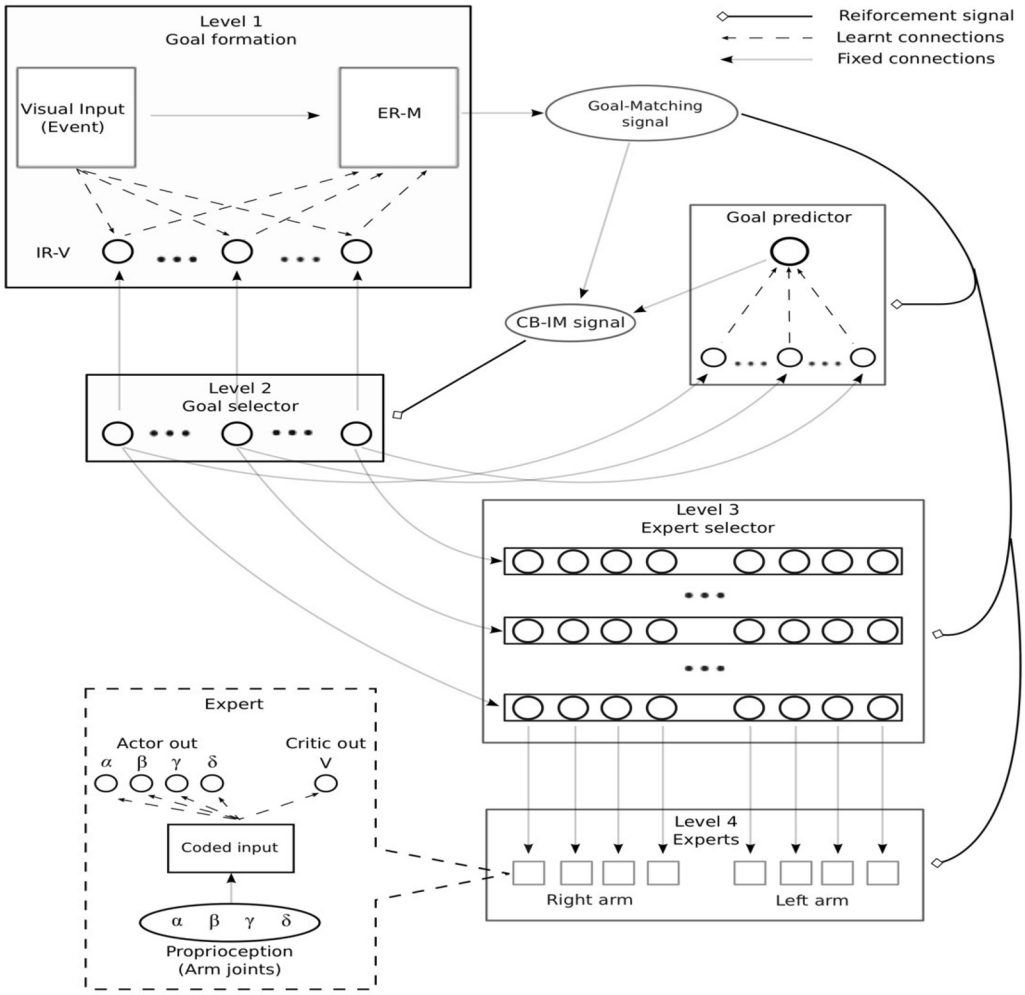


Fig. 3. Four-level hierarchical architecture of GRAIL: 1) goal-formation mechanisms with the IR-V and the ER-M; 2) goal-selector; 3) expert-selector; and 4) experts. The CB-IM signal is also presented in the figure, together with the GM reinforcement signal.

check if the goal is achieved and generate a GM signal. The ER-M has a threshold activation (GM threshold) that can be exceeded only when at least one of its units is activated both by the IR-V input, representing the goal that has been selected, and the input coming from the visual perception of an event. In this way, the threshold can be exceeded only when the event represented in the ER-M is both desired (activation from the IR-V) and happening (activation from the visual input). When the robot discovers a new possible goal it needs several presentations of the same event to modify the weights connecting the IR-V to the ER-M. When the weights have reached sufficiently high value (see Section III-A and the Appendix), we can say that the system has formed a representation of the goal as the GM signal can be triggered. From this moment, if the robot determines the change in the environment that corresponds to the goal it is pursuing, at least one of the ER-M units exceeds the GM threshold and the system auto-generates a signal for the achievement of the goal. On the other hand, if the robot causes an event that is different from the active goal, no unit of the ER-M is able to exceed the threshold and the mechanism generates no GM signal.

The GM signal is used for different purposes: 1) determining the teaching input of the predictor which contributes to generate the CB-IM signal (Section II-D) reinforcing the goal-selector, and determining the reinforcement signal that is used for training both; 2) expert selector; and 3) selected expert.

D. CB-IM Mechanism

The CB-IM signal driving the selection of the units of the goal-selector is the competence-based intrinsic reinforcement signal that we identified in [74] as the most suitable to drive the selection of different goals and the acquisition of the related skills. In particular, the IM signal is the PEI of a predictor that receives as input the output of the goal-selector (the selected goal) and produces an output that can be interpreted as the predicted probability that the event associated to the selected goal will happen. The predictor is trained through a standard delta rule using the achievement of the selected goal as the teaching input (1 for success, 0 otherwise).

E. Overall Functioning of the Architecture

At the beginning of the simulation the robot has no representations stored in the IR-V and the ER-M, so its behavior is completely task-independent: the selection of a unit in the goal-selector has no effect since the system has not discovered

any event yet. This selection does not even affect the selection of the experts as there are no goal-expert associations yet. This implies that the system selects one of the available experts with a flat probability and the robot explores the environment on the basis of this expert and the associated arm. At the beginning of every trial the arm associated with the selected expert is randomly positioned (the positions where the robot touches one of the spheres with its forefinger are excluded). Then, as explained in Section II-B4, the arm is controlled on the basis of the input provided by the position of the joints. The four outputs of the expert are added a noise value (see the Appendix for a detailed description of the noise) and due to the velocity control of the arms these explore the working space randomly since at the beginning of the simulation all the weights inside the experts are set to 0.

When a sphere is touched it lights up, thus generating a change in the environment. The visual input of the event determines the activation of the goal-formation mechanism. As long as the agent has not formed a representation of the goal, no learning signal nor IM signal is generated by the system. The more the robot is able to reach the same object, the more it associates that event to a specific unit in the IR-V, and builds a representation of it in the ER-M. When a goal is properly formed (Sections II-C and III-A), if the robot achieves the selected target it is able to autonomously produce the GM signal (Section II-C) that contributes to the formation of the CB-IM signal (Section II-D) and provides the reinforcement to train the learning of skills.

The process of goal selection is completely autonomous. At the beginning of every trial, on the basis of the CB-IM signal the robot autonomously selects its own goal. Due to the nature of this IM signal, the system is motivated to select those goals where the robot is improving the skill that allow it to determine the related event. When the robot is able to systematically achieve that goal, the IM signal fades away letting the robot free to explore the environment, discover new goals, and learn new skills. Note that here the robot is motivated only to improve its competence (acquiring new skills), so the autonomous selection of the goals is based only on this principle. However, GRAIL could be used also with any other type of motivation that can drive the robot to autonomously select its goals following the different criteria (e.g., energy harvesting or externally assigned tasks).

Regarding learning, GRAIL undergoes five learning processes. Two are associative processes that take place within the goal-formation component: the first allows the IR-V to associate implicit representations of the goals to the event perceived through the visual input; the second associates these implicit representations to the explicit representations in the ER-M, so that the system is able to autonomously recognize, through the GM mechanism, when a desired event is achieved. The other three ones are reinforcement learning processes taking place in the other components of the architecture: based on the intrinsic reinforcement provided by the PEI of the predictor of the achievement of the selected goals, the goal-selector learns to select those goals that have the highest improvement rate; on the basis of the GM signal, the expert-selector learns to select the expert that better allows to cause the event corresponding to the active goal; last, the GM signal is also used to train the selected expert to achieve the desired goal.

III. Experiments and Results

This section first illustrates the functioning of the goal-formation mechanism and the generation of the GM signal (Section III-A), then presents the results of the four experiments. When GRAIL is compared to different systems, details of the architectures with which it is compared are provided in the presentation of the experimental setup.

The first experiment lasts 40 000 trials, while experiment 2 runs for 50 000 trials and experiments 3 and 4 run for 35 000 trials. Each trial ends when the robot touches a sphere, or after a time out of 800 time steps, each lasting 0.05 s. At the beginning of every trial the goal-selector selects a goal. At the beginning of the experiment the units of the goal-selector are not associated with any event. When the goals are formed, the selection of a unit associated with a change in the environment determines which goal the system pursues. Then the selector of the experts determines which expert (and hence which arm) is used to control the robot and learn to cause the event associated with the selected goal. The joints of the selected arm are then randomly initialized and the next trial starts. Every 500 trials the simulations are stopped, learning is switched off, and the performance of the system in all the reaching tasks is tested. More precisely, to this purpose we bypass the goal-selector mechanism by directly activating the units of the goal selector that the robot has associated to the goals. We do this for 100 trials for each unit/goal. The results of these tests are presented as the average performance of the robot (over ten replications) in achieving the different goals in the experiments.

A. Goal Formation and Goal-Matching Signal

As described in Section II-B1, the goal-formation mechanism generates in the ER-M the representations of the events discovered by the system. These representations of the goals are progressively formed if the robot perceives the same event multiple times during the simulation. As shown in Fig. 4, at the first presentations of the event the representation of the goal is still not formed, i.e., the input provided by the IR-V is not able to activate the ER-M higher than the goal-formation threshold (red dotted line in Fig. 4). For this reason, even if the system is able to cause the proper event (here goal 2 of the first experiment), the visual input of that change in the environment is not sufficient to make at least one of the units in the ER-M exceed the GM threshold (green line in Fig. 4).

After some presentations of the same event (18 in this example), the weights connecting the IR-V to the ER-M have been modified to reach a level sufficient to form a representation of the goal. This means that when the system is able to accomplish the selected goal the activation of the ER-M provided by the summed input from the IR-V and the visual input of the event is able to generate the GM signal.

In Fig. 5, a different situation is presented. Here the system has formed a proper representation of the goal [Fig. 5(a)] so

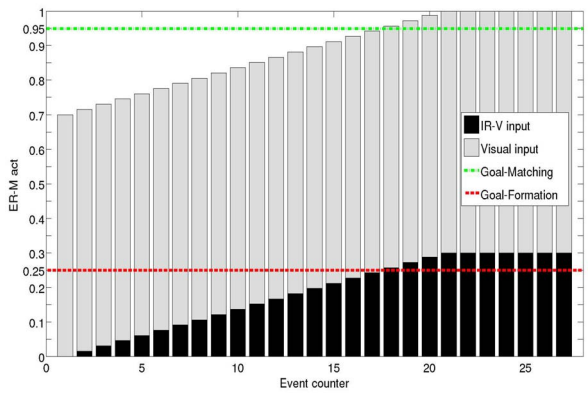


Fig. 4. Process of goal formation (related to goal 2 of experiment 1) and the generation of the GM signal. Data show the activation of 1 unit (related to the selected goal) of the ER-M (*y*-axis) after different presentations of the same event (*x*-axis). The activation of the ER-M is caused by the summed input provided by the IR-V (black) and the visual input of the event (light gray). The production of the GM signal (at least 1 unit exceeds the GM threshold, indicated by the green dotted line) is possible only when the activation caused by the IR-V input exceeds the goal-formation threshold (red dotted line), i.e., when the system has formed a representation of the goal.

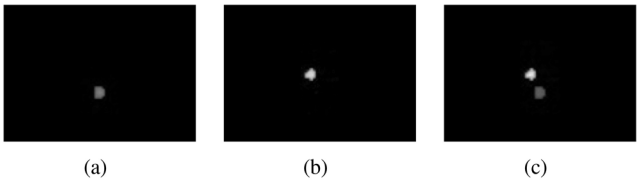


Fig. 5. Process of GM when an event different from the currently pursued goal is experienced. Data refer to the situation where the system (a) has selected a goal (goal 2 of condition 1 as in ER-M activation) but (b) its action determines an event that is different from pursued goal (visual input of the different event generated by the system). (c) Resulting activation of ER-M reflects the difference between the goal that the system is trying to achieve (low activation) and the actual event determined by robot actions (high activation, but not sufficient by itself to trigger a matching signal).

that if the robot causes the proper event the system is able to recognize its achievement and generate the related GM signal. However, the robot now generates an event that is different from the one that it is pursuing. The figure shows the moment when the system is pursuing goal 2 and touches the sphere related to goal 3 [Fig. 5(b)]. The sum on the ER-M resulting from the representation of the selected goal and the actual generated event [Fig. 5(c)] reflects two different patterns corresponding to the two goals (the units of the pattern corresponding to the actual generated event are more active), but none of the two is able to make the units of ER-M exceed the threshold and generate the GM signal.

These results show the ability of the system to gradually form a proper representation of the events associated to different goals so that when the system selects a goal it is autonomously able to "recognize" its achievement and self-generate a GM signal that is used to motivate and train various components of the architecture. Moreover, data of Fig. 5 also show the ability of the robot to discriminate different events when the caused event does not correspond to the goal that the system is pursuing.

B. Experiment 1

1) Experimental Setup: In the first experiment, we test GRAIL in a task where the robot has to learn to reach four different targets all positioned close to the *y*-axis that divides the workspace in left and right [see Fig. 2(a)]. All the objects are reachable by both arms of the robot but it is not known *a priori* which is the best solution to achieve each target, i.e., which arm provides the most efficient way to touch each sphere.

Here we compare the results of the robot controlled by GRAIL to those of our previous systems, the CS and the DS systems. As described in Section II-B, in the CS the goal-selector and the expert-selector are collapsed in a single component with a predefined coupling between goals and computational resources, so that each expert is rigidly associated to a specific goal unit. In the current scenario, as often in real environments, it is not possible to know *a priori* which is the proper solution to solve a task, for example here which arm should be used to achieve each sphere. Thus, in CS the associations between goals and arm-experts have been based on simple spatial correspondences. In particular, considering that in this experiment there are four targets, two positioned on the left and two on the right, the CS has only four experts, two controlling the right arm and two controlling the left arm. The DS, with its three-level decoupled architecture, is able to learn to associate goals and modules to achieve them but it does not have the mechanism necessary to discover the possible goals (so this system lacks the highest level of GRAIL described in Section II-B1). As CS, this system has the goals predefined before the experiment so the goal-selector mechanism is composed of only 4 units, each one standing for a different goal. Similarly to GRAIL, also this DS has eight experts (four for each arm) so that the robot can choose which expert and arm to use to learn the skills related to the different goals: since it is possible that the best solution is to reach all the targets using the same arm, we give the possibility to the system to learn to reach all the spheres with a different expert controlling the same arm.

This experiment will test if a complex architecture such as GRAIL is still able to perform better than a two-level architecture with fixed goal, and goal-expert associations, in a reaching task whose solution is unknown at design time. Moreover, we want to compare the result of DS to those of GRAIL and see if the autonomous discovery of goals slows down the learning process.

Our expectation is that, as in our previous work [72], the DS performs better than the CS. We also expect GRAIL performance to be lowered by two factors: 1) GRAIL has to autonomously discover new goals, to learn to associate the events to the implicit representations in the IR-V and to the explicit representations in the ER-M, while the DS and CS can select goals from the beginning of the experiment and 2) GRAIL has more units in the goal-selector than the other systems (ten versus four) so that the selection process can be slowed down by units not associated to any goal. However, we believe that GRAIL, thanks to its decoupled architecture, is able to perform better (or at least equally) than the CS,

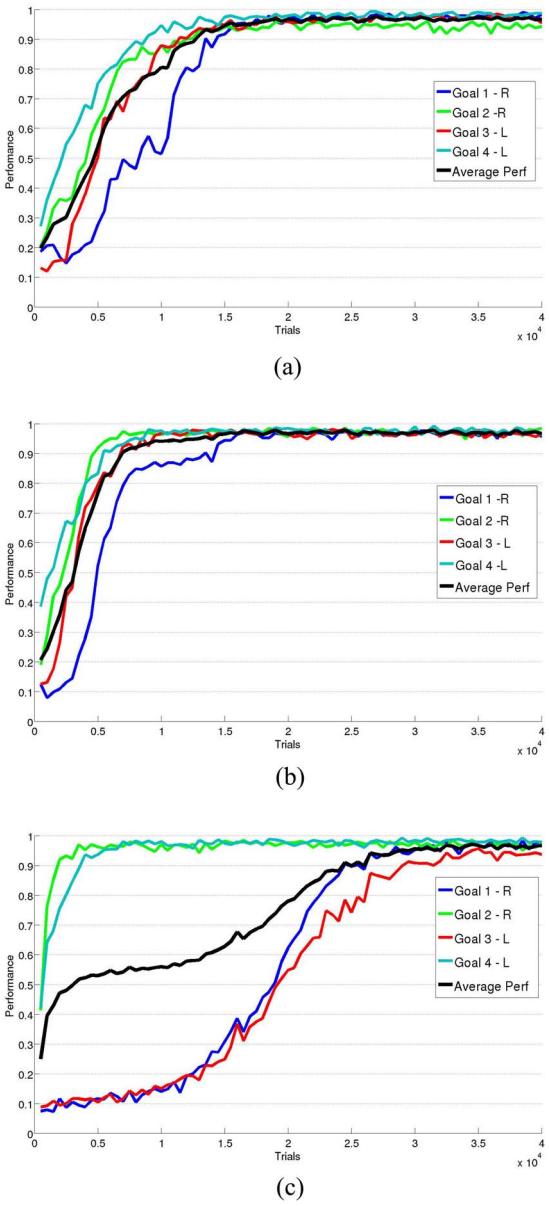


Fig. 6. Performance of the three systems tested in condition 1. Data refer to the averages of ten replications of the experiment. (a) GRAIL. (b) DS. (c) CS. The letters *R* and *L* in the legends refer to the positions of the spheres associated to the goals (right or left) with respect to the *y*-axis that divides the workspace of the robot.

while only experiments can provide a measure of the different performance of DS and GRAIL.

2) Results: The average performance of 10 replications of each experiment are shown in Fig. 6. The CB-IM signal is able to drive all the systems to learn the skills related to the different goals. As expected, both GRAIL [Fig. 6(a)] and DS [Fig. 6(b)] are able to achieve a high performance on the four tasks (∼95%) faster than CS [Fig. 6(c)] which reaches that performance only at the end of the experiment (∼30 000 trials). Although DS is slightly faster (∼12 000 trials) than GRAIL (∼17 000), the difference between the two systems is minimal considering that DS (as well as CS) has all the goals set at design time while GRAIL has to autonomously discover them (as shown in Section III-A).

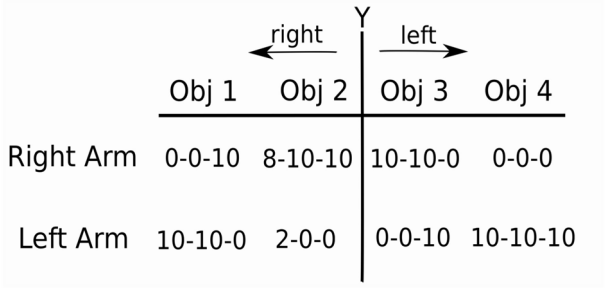


Fig. 7. Number of replications in which GRAIL, DS, and CS (first, second, and third number in each cell, respectively) use the left or right arm to reach each object.

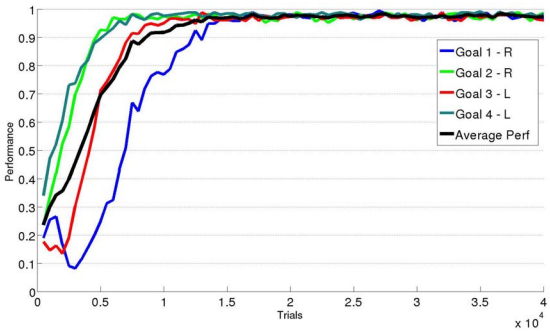


Fig. 8. Performance of GRAIL in the first experimental condition where the goal-selector component of the architecture is composed of 4 units instead of 10. Average data on ten replications of the experiment.

If we look at Fig. 7, we can understand why GRAIL and DS perform better than CS. Data show the arms that the three systems use to achieve the different goals in the ten replications of the experiment. Where all the systems use the same arm (objects 2 and 4) CS is the fastest system since GRAIL and DS have to learn to select the correct arm while CS has goals and experts associated at design time. However, the advantages of being able to autonomously associate computational resources and tasks are evident for those goals for which it is not possible to determine *a priori* which is the best arm to use. The performance of CS is drastically slowed by the learning of goals 1 and 3, while GRAIL and DS autonomously recognize that those objects can be more easily reached with the arm opposite to their position in the workspace and hence take much less time to learn and accomplish those goals.

As shown in Fig. 7, GRAIL learns to achieve the goal related to the second object eight times (on ten replications) with the right arm and two times with the left arm. Since data on the three experimental conditions show that the fastest solution is to reach object 2 with the right arm, this explains why the average performance of GRAIL on goal 2 [see Fig. 6(a)] is slightly lower than the performance on the other goals. However, the lower performance is limited to the two left-arm-solution replications, while performance is high when the suitable arm is selected.

This is confirmed by the results of a test in which we provide the IR-V and the goal-selector of GRAIL with only 4 units rather than 10 (Fig. 8). In this condition the system reaches high performance with all objects similarly to DS

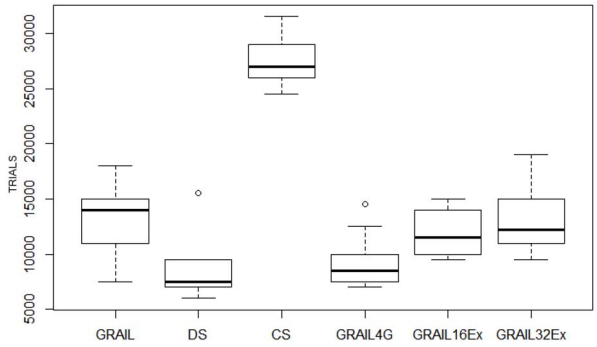


Fig. 9. Box plots of the performance of the tested conditions (ten replications each) with respect to the number of trials needed to achieve an average performance of 95% on the four tasks of experiment 1. In addition to GRAIL, DS, and CS we also test three different versions of GRAIL: with only four goal units (GRAIL4G), with 16 experts (GRAIL16Ex), and with 32 experts (GRAIL32Ex). The whiskers of the box plots indicate the minimum and maximum values with the exception of the outliers when present.

(∼95% in ∼12 000 trials). Moreover, these results show that the process of goal formation per se does not slow down the learning process of the system: rather, it was only the higher number of goal units that slowed down the learning of GRAIL in the original experiment.

To test the scalability of GRAIL with respect to the number of its experts, we perform two further tests (ten replications each) where GRAIL is provided with 16 and 32 experts, respectively. The performances of the system in these conditions are reported in Fig. 9 and show that increasing the number of experts does not impair the velocity of the system with respect to the version of GRAIL with only eight experts.

To provide a statistical analysis of these results, we measured the time (number of trials) needed by the robot to achieve an average performance of 95% in the four tasks (all the conditions are able to achieve a ∼100% performance so we considered the achievement of 95% a good measure of the learning success). The box plots of the different conditions are presented in Fig. 9. A one-way ANOVA reveals significant differences among the tested conditions ($F(5, 54) = 72.48$, $p < 0.0001$). We then run a *post-hoc* test to support the results. In particular, we find significant differences between DS and CS ($p < 0.001$), GRAIL and CS ($p < 0.001$), and GRAIL and DS ($p < 0.01$). As previously underlined, no significant difference was found between GRAIL with four goal units (GRAIL4G) and DS. Moreover, no significant differences were found between the version of GRAIL with eight experts and those with 16 (GRAIL16Ex) and 32 (GRAIL32Ex) experts, thus confirming the capability of GRAIL to scale up with respect to the number of computational resources provided to the system.

C. Experiment 2

1) Experimental Setup: In the second experiment we want to test the properties provided to GRAIL by the goal-formation mechanisms (Section II-B1). We put the robot in a new scenario that is intended to mimic (although in a very simple way) some of the situations that an artificial agent could encounter in a real environment, where actions may turn out to have different effects in different moments or where new possible goals may appear during the learning.

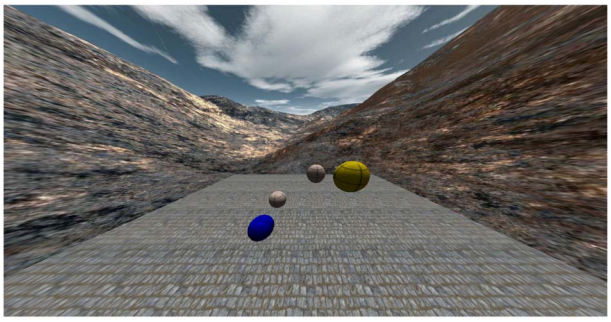


Fig. 10. Experiment 2 seen from the robot perspective. The two gray spheres are "normal" targets present from the beginning of the simulation. The yellow sphere is present from the beginning of the simulation but becomes active (it can be lighten up by the robot touching it with its forefingers) after 15 000 trials. The blue sphere is not present at the beginning of the simulation and appears after 25 000 trials. Yellow and blue colors are only used in this figure to identify the different spheres.

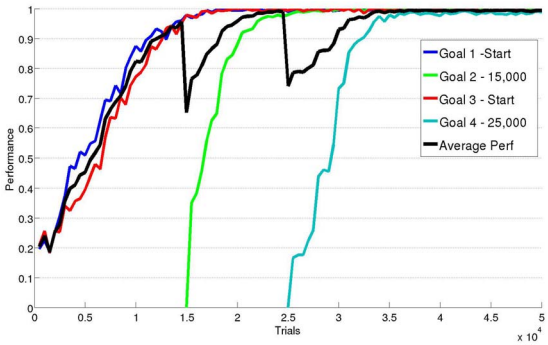


Fig. 11. Performance of GRAIL in experiment 2. The legend indicates the moment from when the goal can be discovered: from the beginning of the experiment (start) or after a certain amount of trials. Average data on ten replications of the experiment.

The task still consists in learning to light up different spheres but this time the goals are not all present from the beginning. When the experiment starts, the robot finds three spheres in its workspace but only 2 of them can be lighten up (Fig. 10, gray spheres). Later (after 15 000 trials) a third sphere (Fig. 10, yellow sphere) becomes active providing a new possible event/goal for the robot. Moreover, at a certain point of the experiment (after 25 000 trials) a fourth sphere (Fig. 10, blue sphere) is introduced in the environment adding a further goal to be discovered.

The expectation is that GRAIL is able to manage this complex situation, thus showing that a system that is able to autonomously discover new goals has an improved versatility and adaptation. This provides a clear improvement with respect to the previous architecture (the DS system [72]) which is not able to manage situations where goals are not known at design time.

2) Results: Fig. 11 shows the performance of GRAIL in experiment 2 (average performance of ten replications of the experiment). In the first 15 000 trials only goals 1 and 3 can be achieved, and the robot is able to learn the related skills. After 15 000 trials also goal 2 is available. The average performance

Fig. 12. Experiment 3 seen from the robot perspective. The yellow sphere activates independently of the robot actions with a probability of 0.15. The red sphere is outside the workspace of the robot and activates independently of its actions with a probability of 0.20. The other spheres activate when touched by the robot as in previous experiments. Yellow and red colors are only used in this figure to identify the different spheres.

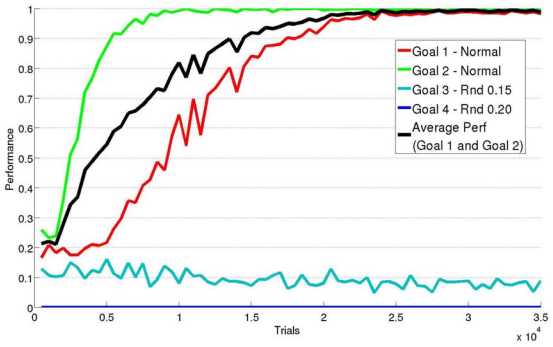


Fig. 13. Performance of GRAIL in experiment 3. The legend indicates when the goals are normal or independent of the robot activity (random goals, rnd). Since in this scenario the robot can only learn two skills, the average performance of the system is calculated on the achievement of the two goals depending on robot activity.

decreases as now it is computed also on the new task which has to be learned, but in few trials (∼10 000) the robot is able to discover the new event, form a new goal related to it, and learn the related skill. After 25 000 trials a new sphere is added. After a second decrease of the average performance (now calculated over four tasks) the robot is able to form the last goal and train the last skill and achieve a high performance (close to 100%) in all the tasks.

As expected, GRAIL is able to manage such a complex situation discovering the new goals as they become active or appear in the environment, providing the goal-selector with new associated nodes that can be selected, on the basis of the CB-IM signal, and guide the learning of the related skills.

D. Experiment 3

1) Experimental Setup: GRAIL is developed to recognize and form representations of changes in the environment so to form a set of possible goals that the robot can select to learn the skills necessary to achieve them. However, the mechanism of goal discovery implemented in GRAIL could have problems in real environments where many changes happen without the intervention of the robot. In particular, the agent could be distracted by these changes and focus on learning to cause them although they do not depend on its activity.

To evaluate if GRAIL is able to cope with these situations, we test the robot in a setup where there are four spheres (Fig. 12): two spheres are activated by the robot touch as in previous experiments; the other two spheres, one positioned within the workspace of the robot and the other one outside (yellow and red sphere in Fig. 12, respectively), are not affected by the robot activity and have a probability of, respectively, 0.15 and 0.20 of activating during a trial.

The expectation is that GRAIL forms representations of the random events and associates them to the goal-units in the goal-selector. At the same time, the system should be able to avoid the distraction provided by the random events and learn the skills that allow the activation of the two spheres that can be affected by its actions. The reason is that GRAIL is driven by a CB-IM signal based on the improvement of competence: since there cannot be any improvement in causing the activation of the random spheres, the robot should focus on learning the skills related to the other two spheres.

2) Results: Fig. 13 presents the average performance of GRAIL in 10 replications of experiment 3 with respect to the goals stored by the goal-formation mechanisms. As expected, the system has formed also representations of the random events (for this reason they are included in Fig. 13) although they do not depend on the activity of the robot.

Notwithstanding the formation of "distracting" goals, the learning process of GRAIL is not impaired: as we can see from the performance on goals 1 and 2 the robot is able to learn to reach for the related spheres achieving a high average performance on the two learnable tasks. Instead, GRAIL does not learn to reach for the sphere related to goal 3 which, although randomly activated, is positioned within the workspace of the robot (the sphere associated to goal 4 is positioned outside the workspace and cannot be touched).

These results confirm that the CB-IM signal based on the improvement of the competence of the robot is able to guide the learning of skills even in environments where there are events that are not dependent on the robot activity. Indeed, the PEI CB-IMs motivates the system to train the skills that can be improved whereas those related to random events are not pursued by the robot.

E. Experiment 4

1) Experimental Setup: The last experiment (Fig. 14) tests if GRAIL is able to cope with events that have a stochastic response to the actions of the robot. Since the system is driven by an IM signal related to the competence in the skills that it is learning, there might be the possibility that if the agent is involved in performing an action that has a stochastic effect it gets stuck in trying to improve an ability which cannot be improved.

For this reason we test GRAIL in a new experimental setup where there are 3 spheres: one is a normal sphere that is activated by the robot touch (Fig. 14, gray sphere); another is a stochastic sphere that is activated with a probability of 0.75 when touched (Fig. 14, red sphere); the last one is a sphere

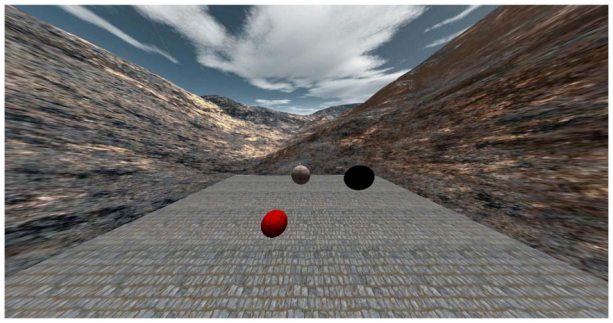


Fig. 14. Experiment 4 seen from the robot perspective. The black sphere activates only the first ten times it is touched by the robot. The red sphere activates with a probability of 0.75 when the robot reaches it. The last sphere is a normal object that lights up every time the robot touches it with its forefingers. Red and black colors are only used in this figure to identify the different spheres.

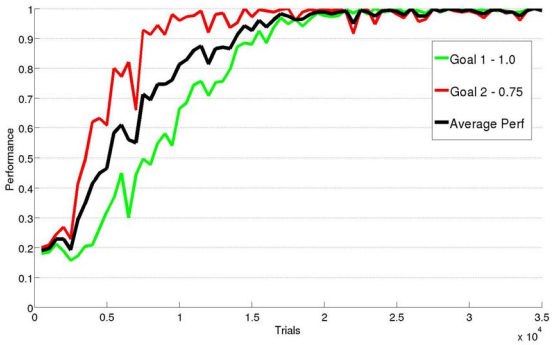


Fig. 15. Experiment 4: average performance of ten replications of the experiment of the system driven by the PEI CB-IM signal. Data refers to the two goals achievable by the system. The numbers next to the two goals in the legend refer to the probability of the two goals to occur when the robot touches the related sphere.

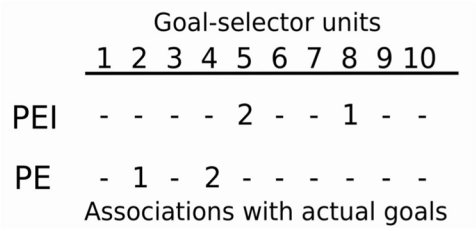


Fig. 16. Experiment 4: associations of the 10 units of the goal-selector with the actual goals. Only the two goals that the system can perform have been associated to the units of the goal-selector. No unit has been associated to the event related to the sphere that can be lighten up only for the first ten times that it is touched. Data refer to two representative replications of the system driven by the PEI signal and the PE signal.

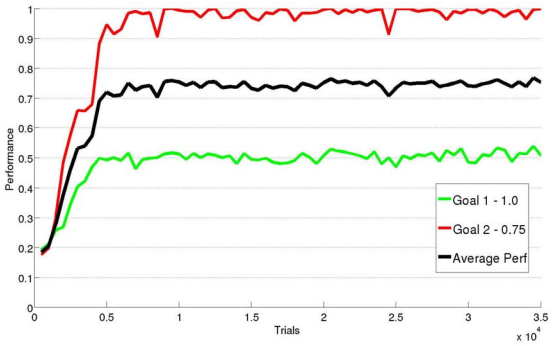


Fig. 17. Average performance on ten replications of the experiment of the system driven by the PE signal. Same data as Fig. 15.

reachable by the robot but that lights up only for the first ten times that is touched by the robot (Fig. 14, black sphere).

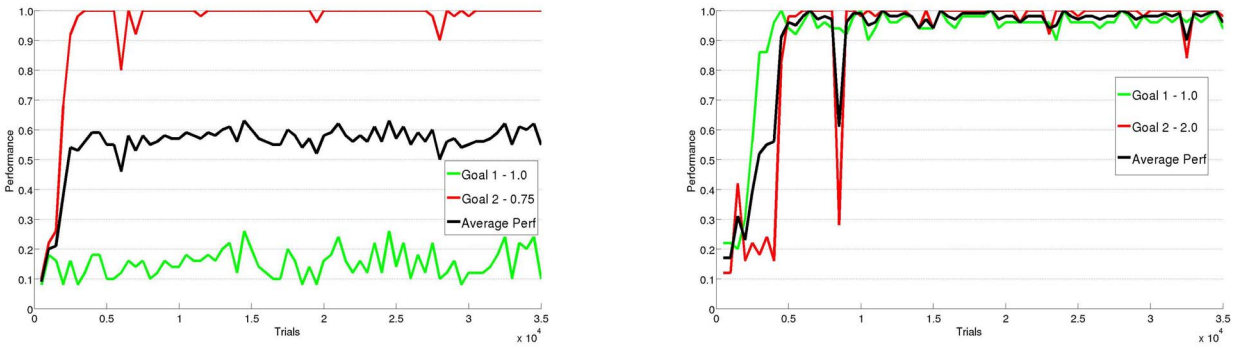
To demonstrate the advantage of using a PEI signal, in this experiment we compare the results of GRAIL with those of an identical system whose intrinsic reinforcement signal is determined by the simple PE of the predictor of goal achievement. The importance of this test resides in the fact that in our previous works [71], [74] we found that in nonstochastic environments the PE generates a stronger and more stable signal for guiding skill acquisition with respect to PEI, so the advantage of PEI over PE in the particular setup used here should not be given for guaranteed. For this reason, we wanted to test both mechanisms in the current setup and also (and more importantly) to verify their behavior when working in concert with the other various mechanisms of the architecture, including learning mechanisms that have differential learning speeds.

Our hypothesis is that the CB-IMs signal based on the PEI is able to manage the uncertainty of this setup, as suggested by [60] and [78]. In particular, we expect that the robot is able to learn to achieve not only the normal sphere but also the stochastic sphere without getting stuck on it, differently from the PE-driven system that we expect to be unable to cope with this situation. Moreover, with the third sphere we check whether, as desired, only events that occur a certain number of times are able to become goals for the system. Indeed, a sphere that only lights up few times should not be stored with the other representations of goals as it represents a transient action-outcome contingency.

2) Results: Fig. 15 shows the average performance of ten replications of the experiment. The robot is able to properly learn to activate the two spheres that depend on its actions (objects 1 and 2). Although there is a stochastic target that can be lighten up only with a probability of 0.75, the robot is able to learn the related skill (goal 2 in Fig. 15) and then focus on discovering and learn the other task (goal 1). Moreover, if we look at Fig. 16 we can see that the system does not assign any goal unit to the event that only happens few times. This guarantees that the system only forms representations of events that have a minimum reliability.

The problem that such a stochastic environment can give to a system driven by an IM signal is clear when we test GRAIL using a PE signal instead of the PEI signal. Also in this case the system does not assign any of the units of the goal-selector to the event that only occurs the first few times at the beginning of the experiment (Fig. 16). However, if we look at the performance of the system driven by the PE signal (Fig. 17) we can see that the robot may get stuck in trying to improve skills that cannot be further improved: the agent is able to properly learn the skill related to the stochastic goal, while it only achieves an average performance of ∼50% (on average in 10 replications) on the skill related to the "normal" sphere.

The reason is that in replications where the robot focusses first on the stochastic goal [Fig. 18(a)], even after it has learned



(a) (b)

Fig. 18. (a) Performance of the PE system in a replication where it first focus on the stochastic goal. (b) Performance of the PE system in a replication where it first focus on the normal goal.

the related skill it continues to receive a PE signal that reinforces the selection of the associated unit in the goal-selector. This is due to the fact that for the stochastic goal the predictor learns to generate, on average, a prediction of 0.75, which makes the system keep receiving a reinforcement (PE) of about 0.25 every time the goal is achieved, thus the goal-selector never stops to select that goal. Differently, when the robot first focusses on goal 1 [Fig. 18(b)], it can learn both skills since the first goal has not a stochastic activation and, when the related skill has been learned, the predictor is able to cancel the related IM signal thus allowing the agent to switch to learn the skill related to the second goal (which eventually has been already discovered). Instead, when the PEI signal is used the predictor generates a prediction close to 0.75 as in the PE system, but this time the intrinsic reinforcement fades to 0 when the predictor is not able to improve its prediction anymore.

IV. Discussion

In this paper, we presented GRAIL. GRAIL is an autonomous system that, to the best of our knowledge, for the first time is able to assemble in an integrated architecture different mechanisms that are necessary to have truly autonomous open-ended learning robots. In particular, with its four levels GRAIL is able to autonomously: 1) discover changes in the environment that the robot can cause through its own action; 2) form representations of the goals corresponding to those changes; 3) select one goal to pursue at each moment on the basis of the CB-IM signal; 4) select suitable computational resources to achieve the selected goal; 5) monitor the achievement of the selected goal; and 6) self-generate a learning signal when the selected goal is successfully achieved.

We tested GRAIL in a 3-D, 4 DOFs task with a two-armed simulated iCub robot, where the robot has to reach for different targets. In particular, we performed four experiments to test different capabilities of the system.

In the first experiment (Section III-B), we compared GRAIL with two different systems, a three-level decoupled architecture (DS) and a two-level coupled architecture (CS): the better performance of GRAIL and DS with respect to CS confirms the importance of providing the robot at least with a three-level decoupled architecture that allows the system to both select its goals and search for the better computational resources to achieve them.

In the second experiment (Section III-C), we focussed on the importance of the higher level of GRAIL that allows the robot to autonomously discover new goals without any intervention by the programmer. This is far beyond what the three-level DS architecture can do: indeed, in a scenario where possible goals are not known at design time, only a system like GRAIL is able to discover new events and adapt to a dynamic environment where goals can appear or modify during time.

In the third experiment (Section III-D), we tested GRAIL in a setup where some events happen independently of the robot activity. Although the system forms goal representations of these events, the learning process of GRAIL is not impaired thanks to the CB-IM signal based on competence improvement that motivates the robot to focus only on those skills related to the events that can be caused by the robot.

In the fourth experiment (Section III-E), we tested GRAIL ability to cope with environments where the actions of the robot have a stochastic effect on the world. In particular, we tested the importance of using an IM signal based on the PEI that, differently from a signal based on the simple PE, disappears when the system has reached maximum competence, thus preventing the robot from getting stuck in trying to improve skills that cannot be improved.

GRAIL is linked to other works in the field of autonomous learning that underline the importance of goals for the autonomous development of artificial agents and in particular for skills learning. At the same time, our approach to goals differs from previous systems proposed in the literature.

The option framework [85] implements goals as the termination condition of an option policy. However, the works in this framework have mainly focused on discrete state and action domains [12] possibly with pre-established goals [82]. Differently from this, GRAIL has the ability to autonomously discover goals and learn skills without any intervention of the programmer in continuous domains. Some efforts have been made toward this direction in the field of hierarchical reinforcement learning [13] but some of these works [2], [41], [48] focus on searching for subgoals on the basis of some sort of externally given tasks. In [42] the system

is able to perform a sort of representation learning, but starting from preassigned abstraction that comprehend all possible object-effector associations. Other systems are able to set their own goals without being externally assigned any task but some of them focus on the acquisition of low-level motor skills relying on suboptimal KB-IMs [56] (see [71] and [74] for a comparison between different typologies of IM signals), or they are implemented in disembodied agents [49], [90].

Merrick [49] presented an interesting system where CB-IMs are used to guide a two-level architecture in selecting its goals and training the related options. The system enlarges the repertoire of possible actions through a novelty-seeking "introspection" level that is able to improve the ability of the system in focusing on more complex goals and efficiently learning the related skills, providing also a strategy to delete those skills (options) that had become useless. The setup used in that work is a discrete grid-world scenario with a virtual agent that learns to select its actions using a (growing) repertoire of "tools" that turn out to be useful in specific states. This is different from our setup, where we use a 3-D environment with a 2-arm four DOFs simulated robot that has not only to select its own goals but also to learn how to perform the low-level actions needed to control the robot arms to achieve the goals.

The goal-babbling framework (Section I-B) has underlined the importance of goals in the optimisation of learning processes in high-dimensional action spaces involving redundant robot controllers [11], [68], shifting the exploration of the artificial agents from the motor-space to the task-space. These systems are even able to autonomously discover and set new goals. However, in these works goals are considered as states related only to the body of the robot. This kind of goals can drive a robot to learn to control its own body and form a model of it, but they cannot drive the acquisition of skills related to changes in the external environment, which is what we are interested in here.

In [66], the proposed system, using goal babbling, is able to autonomously identify an object as a salient element of the environment, but the reward function is designed as the distance between the robot effector and the object: in this way even if there are no effects in the environment, the agent obtains a reinforcement signal that guides its actions in a reaching task that is in this way supervised by the programmers. Moreover, if the interaction with an object has no effects, the agent is still reinforced to reach for it. Our idea is that the reward function should not be connected to the saliency of an object per se, but to the fact that the robot actions causes relevant changes in the environment. Importantly for the autonomy of the system, note that, although simple, the mechanisms of event-detection and goal-formation implemented in GRAIL are general and do not depend on the particular events scheduled for this specific setup. Thus, if a new action-triggered event is introduced in the environment, the system, without any change, would be already able to detect it and form a new goal corresponding to it.

In this respect, our approach can be considered as complementary to goal babbling. We use goal discovery and goal selection to learn low-level skills (here reaching) through motor babbling exploration. The goal babbling approach seems to be more efficient to learn skills at such a low level and so it might be employed in our architecture. Once learned, those skills could represent the motor building-blocks to acquire higher-level skills, identified through an architecture such as GRAIL that discovers (and sets) its goals on the basis of the events that modify the environment external to the robot.

In this implementation of GRAIL we do not address the issue of abstraction in relation to goals (see [66]). In particular, a key enhancement of GRAIL would be to endow it with the capacity to form abstract goals, namely goals encompassing a set of the environmental states having in common some relevant features.

An important feature of GRAIL (shared with our previous DS architecture [72]) is the decoupling between the selection of the goals and the selection of the computational resources (here the experts) with which the system tries to achieve the selected goals. The importance of this ability is shown in experiment 1 (Section III-B) where only a DS is shown to be able to properly discover the most suitable strategy to achieve a task. However, the setup used here offers a limited choice with respect to both the effectors (the robot can interact with the world only with the two arms) and the computational resources (all the experts are identical). In future works it would be interesting to: 1) test GRAIL in more complex scenarios and provide the robot with a wider range of effectors (e.g., the control of fingers); 2) implement different experts that vary in the composition of their input, for example using different preprocessing features; and 3) implement experts that differ in their internal structure (e.g., the number of units). These differentiations of the experts provide a wider choice of resources to the selection mechanism of the architecture. Since GRAIL is able to autonomously select the most efficient experts, increasing the variability of the experts and providing the system with wider choices can result in an improvement in learning speed and performance.

GRAIL shares also some features with previous systems that used actor-critic experts within a two-level hierarchical architecture, but it has also important differences with respect to them. A first system [3] shares with GRAIL the use of actor-critic experts. However, these experts are used to solve different parts of a whole complex navigation task that is segmented by an "expert selector" neural network on the basis of the specialisation and capacity of experts to solve those parts (these mechanisms have been substantially improved in the following works, see [22], [88], [89]). This challenge has also been faced by a similar system [36] that used actor-critic experts to act in the different parts of the whole task but showed the advantages of using unsupervised learning processes (a Kohonen network) to segment the tasks in subparts. All these systems, however, can only solve extrinsically rewarded tasks and do not have the capacity of self-generating goals and pursuing them through IM learning signals as GRAIL does.

A limitation of GRAIL resides in the implementation of its lower level components. The experts controlling the arms of the robot are implemented as classical actor-critic modules (e.g., implemented with an expanded code of the input and a

learning linear component) which are known to be suboptimal in robotics. In particular, the learning algorithms used to train those implementation of the actor-critic are considerably slow to be used with real robots. A solution to this problem can rely on the use of other parameterized models to encode the policies, for example dynamic movement primitives that are dynamical models capable of producing whole discrete or rhythmic stable trajectories (see [33]), and efficient policy search algorithm to search the parameters of such policies (see [38]).

In the current implementation of GRAIL the goal-formation process does not affect the selection of motor actions that is reinforced (motivated) only after the actual formation of the goal and the consequent generation of the matching signal. This is due to the use of a reinforcement signal that is determined only by the achievement of the goals and by CB-IMs. To cope with this limitation and speed up the learning process, in future works we could use two different solutions. The first one does not require a modification of the architecture of the system: we could simply speed-up the goal-formation process by modifying the learning rates that determine the identification of a new goal in the IR-V and the formation of the related representation on the ER-M. A second solution could rely on providing GRAIL with both CB-IMs and KB-IMs, where the latter could play either or both of the following two roles: 1) identify which events are unexpected, so that the goal-formation mechanism will be activated not for every event, but only for those changes that are still novel or unpredicted by the agent and 2) through signals based on these events, provide an "early" intrinsic reinforcement that is able to modify the behavior of the agent before the actual formation of the goal, biasing the repetition of those actions that determined interesting events.

In an open-ended, intrinsically motivated architecture such as GRAIL, a possible problem could be the generation of an overwhelming large number of goals. Not all the effects that an agent can produce in the environment are necessarily useful for the formation of adaptive actions. While this is not a problem affecting robots tested in laboratory-like scenarios, this is very likely to occur in real environments. For this reason, future autonomous goal-discovering architectures should adopt strategies to limit the formation of new goals and/or to eliminate previously formed ones. Biases to goal formation can be provided both by the system itself and by external users: on the one hand, the system can form general categories (or structures) of similar goals with proven value (e.g., goals that provide high control over the world or goals that allow the achievement of some extrinsic rewards) and prioritise the formation of new goals that fit those categories; on the other hand, if robots are used for particular tasks or in particular situations, human programmers could bias the value system of the architecture so as to privilege the interactions with some particular objects or locations.

Moreover, in real environments many little changes continuously happen due to noise: these minor changes would be considered by the current implementation of GRAIL as events and would cause goal proliferation. In this paper, we were not interested in building a sophisticated event-detector, but in future implementations of the system, especially if real robots are used, we will use suitable filters that would prevent the system from considering as events all minor changes caused by noise.

A further limitation of GRAIL dwells in the visual input and attentional mechanisms, which are very simple and underexploited. Indeed, they could be exploited to improve two different aspects of the system. First, it could enhance the exploitation and learning of skills. In particular, the visual input could be sent to the controllers: the possibility to learn to reach where the eye is foveating [32], [75] would allow the robot to reuse the skills learned with a certain object to interact with other objects located in different positions. Second, vision and attention could support the discovery of new goals: using visual processing techniques such as object recognition [44], [86] and active vision [9], [59] strategies the system could find where objects are located in the space, and also identify novel items between them, so as to focus its exploration in those parts of the environment to speed up the discovery of interesting events.

An element that can influence the open-ended learning nature of the system is the complexity of the goal representations. In the scenarios presented in this paper the events are simple as they correspond to spheres that change their color in one simulation step. If the goal-images that the system has to store are composed of more complex patterns or variations involving several time steps, the process of goal formation would need more sophisticated goal detection component. For example, the identification of interesting events could be provided by more sophisticated algorithms such as slow features analysis (SFA) [39], [94], that is able to identify complex events from a visual stream input (see [40] for an example where SFA and IMs are integrated to foster skill acquisition).

Finally, GRAIL is developed to focus on learning very simple skills such as reaching. However, a true challenge for autonomous robotics, beyond the autonomous discovery and selection of goals, is to build artificial systems able to learn and actuate a true hierarchy of different (and complex) skills, possibly exploiting goals that can recall each other or forming higher level goals from the connection of previously discovered ones. Some efforts have been attempted toward this direction [7], but many of them remain at an abstract level of implementation [13] and others only now start to exploit the power of goals to implement these processes [83].

APPENDIX

COMPUTATIONAL DETAILS OF THE EXPERIMENTS

This appendix provides the details necessary to reproduce GRAIL and the simulations described in this paper. The visual input of GRAIL consists in the perception of changes in the environment (events) provided by the camera of the right eye of the robot. The camera input is a 320×240 RGB pixel image downsampled into an 80×60 RGB pixel image. The events (in our experimental setup taking place when a sphere lights up) are identified by an 80×60 binary image (black and white) obtained subtracting pixel by pixel two consecutive frames. When an event occurs (i.e., when the binary image has at

least one activated pixel), the map is normalized (norm equal to 1) and given to the WTA competitive network as input. To prevent the robot from being "distracted" by its own actuators, we detect the arms and the hands based on a blue color not used for other elements of the setup; in particular, when such color is detected in pixels of either one of the two images used for change detection, we exclude the detection of change in correspondence to such pixels.

Each input unit i is connected with all the 10 units of the vector composing the output of the WTA network, the IR-V. Each weight w_{ji} linking input unit i to output unit j of IR-V is initialized at the beginning of each experiment with a random value chosen in [0, 0.1] and then each set of connection entering one output unit j are normalized to 1. The activation of each output unit j is computed as the weighted sum of the input units. The connection weights of the winning unit j, and only this, are modified as follows:

$$\Delta w_{ji} = \eta \; x_i \tag{1}$$

where w_{ji} is the weight linking input unit i to the winning unit j, η is the learning rate of the WTA network set to 0.3 and x_i is the activation of input unit i. After the modification, the set of weights projecting to j from every input units is normalized to 1.

The same visual input of the event projects not only to the WTA network but also, with fixed one-to-one connections set to 0.7, to the ER-M. The activations of the IR-V and the ER-M determine the modification of the connections projecting from each unit of IR-V to all the units of ER-M. The connection weight v_{ji} connecting IR-V unit i to ER-M unit j is updated through a Hebbian rule (with decay and postsynaptic gating [28])

$$\Delta v_{ji} = \eta_{hb} x_j (x_i - \nu) \tag{2}$$

where η_{hb} is the learning rate set to 0.08, x_j is the activation of ER-M unit j, x_i is the activation of IR-V unit i (1 for the winning unit of the WTA network, 0 for the others), and ν is a value set to $(1/n)$ where n is the number of the units of IR-V (here 10).

The weights connecting IR-V to ER-M are set to 0 at the beginning of the each simulation and their maximum value is set to 0.3. When a unit is selected by the goal-selector (Section II-B2), the information is sent to the IR-V and the active unit determines the activation of the ER-M. The units in the ER-M have a GM threshold activation set to 0.95: if at least one of the units exceeds that threshold, the system generates a signal for the achievement of the goal (the GM signal).

The goal-selector comprises 10 units. At every time step it determines through a softmax selection rule a winning unit. The probability of unit k to be selected (p_k) is

$$p_k = \frac{\exp\left(\frac{Q_k}{\tau}\right)}{\sum_{i=0}^{n} \exp\left(\frac{Q_n}{\tau}\right)} \tag{3}$$

where Q_k is the value of unit k and τ is the softmax temperature, set to 0.008, that regulates the stochasticity of the selection. In experiment 4 (Section III-E), in which we tested GRAIL using an IM signal based on the PE, τ was set to 0.01 (the different temperatures comes from heuristics used in previous works [72], [74] to determine the best values for PE and PEI conditions). At time t the value of the selected unit Q_k^t, representing the motivation to select the corresponding goal, is updated through an EMA of the intrinsic reinforcement (ir) generated for obtaining the goal associated to that specific unit

$$Q_k^t = Q_k^{t-1} + \alpha\left(\text{ir} - Q_k^{t-1}\right) \tag{4}$$

where α is a smoothing factor set to 0.35.

The activation of the selected unit in the expert-selector is updated through an EMA (4), with smoothing factor set to 0.35, based on the reward obtained to achieve the selected goal (1 for success, 0 otherwise). A softmax selection rule (3) is then used to determine the winning unit of the expert-selector (temperature is set to 0.05).

The input to the selected expert are the angles of the actuated joints of the related arm (α, β, γ, δ). This input is encoded with Gaussian radial basis functions [63] with centres on the equally distributed vertexes of a 4-D grid having five elements per dimension (the 5 units cover the range of one arm joint)

$$y_i = e^{-\sum_d \left(\frac{(c_d - c_{id})^2}{2\sigma_d^2}\right)} \tag{5}$$

where y_i is the activation of feature unit i, c_d is the input value of dimension d, c_{id} is the preferred value of unit i with respect to dimension d, and σ_d^2 is the width of the Gaussian along dimension d (widths are parameterized so that when an input is equidistant, along a dimension, to two contiguous units, the activation of the latter ones is 0.5).

Each expert is a neural-network implementation of the actor-critic model [15] adapted to work with continuous states and action spaces [25], [76]. The value produced by the critic of each expert (V) is calculated as a linear combination of the weighted sum of the input units

$$V = \sum_{i}^{N} y_i u_i + b_V \tag{6}$$

where u_i is the weight projecting from input unit i and b_V is the bias. The activation of the four output units of the actor is determined by a logistic transfer function

$$o_j = \Phi\left(b_j + \sum_{i}^{N} u_{ji} y_i\right) \quad \Phi(x) = \frac{1}{1 + e^{-x}} \tag{7}$$

where b_j is the bias of output unit j, N is the number of input units, y_i is the activation of input unit i, and u_{ji} is the weight of the connection linking units i to j.

Each motor command signal o_j^n is determined by adding noise to the activation of the relative output o_j. Since the controller of the robot modifies the desired velocity of the joints progressively, white noise would determine extremely little movements and the arm of the robot would get stuck in a small region of the joints space. For this reason, as in [25], the noise (n) that is added to the output of the actor of the active expert is generated with a normal Gaussian distribution with average

0 and standard deviation (s) 2.0 and passed through an EMA with a smoothing factor set to 0.08. To help the system to manage the exploration/exploitation problem [84], in particular to reduce the time spent by the experts to reach the goals when their competence improves, we implemented an algorithm that allows the system to self-modulate the noise n. In particular, the s of each expert decreases with a "noise-decrease parameters" (d) determined by an EMA (with smoothing factor set to 0.0005) of the success of the expert in achieving the goal for which it has been selected (1 for success, 0 otherwise). The s of the selected expert e (and only this) at trial T (S_{eT}) is updated as follows:

$$S_{eT} = s(1 - d). \tag{8}$$

The actual motor commands are then generated as follows:

$$o_j^n = o_j + n \tag{9}$$

where the resulting commands are limited in [0; 1] and then remapped to the velocity range of the respective joints of the robot determining the applied velocity ($\dot{\alpha}$, $\dot{\beta}$, $\dot{\gamma}$, $\dot{\delta}$).

The expert selected to control the robot in the current trial is trained through a TD reinforcement learning algorithm [84]. The TD-error (δ) is computed as

$$\delta = (r_t + \gamma_k V_t) - V_{t-1} \tag{10}$$

where r_t is the reinforcement at time step t, V_t is the evaluation of the critic at time step t, and γ is a discount factor set to 0.99. The reinforcement is 1 when the robot achieves the selected goal, 0 otherwise. The weight u_i of the critic input unit i of the selected expert is updated as usual

$$\Delta u_i = \eta_c \delta y_i \tag{11}$$

where η_c is the learning rate, set to 0.02. The weights of the actor of the selected expert are updated as follows:

$$\Delta u_{ji}^a = \eta_a \delta \Big(o_j^n - o_j\Big)\big(o_j(1 - o_j)\big)y_i \tag{12}$$

where η_a is the learning rate, set to 0.4, $o_j^n - o_j$ is the difference between the control signal executed by the system (determined by adding noise) and the one produced by the controller, and $o_j(1 - o_j)$ is the derivative of the logistic function.

The IM signal provided to the goal-selector is determined by the PEI of a predictor that receives as input the output of the goal-selector (encoded in a 10-elements binary vector, where 10 is the number of the different units) and that produces as output a prediction (ranging in [0, 1]) of the probability of achieving the goal associated to the current input. The training of the predictor is based on a standard delta rule where the teaching input (g) is the binary encoding of the achievement of the selected goal

$$\Delta p_i = \eta_p (g - p_i) \tag{13}$$

where p_i is the prediction given the input and η_p is the learning rate of the predictor set to 0.05.

The PEI is calculated as the difference between two averages of absolute PEs. Each average is calculated over a period T of 40 selections (related to the same goal), so the two averages cover a period of 80 selections going backward from the current selection into the past. In detail, at time t, the PEI is calculated as follows:

$$\mathrm{PEI}_t = \frac{\sum_{i=t-(2T-1)}^{t-T} |\mathrm{PE}_i|}{T} - \frac{\sum_{i=t-(T-1)}^{t} |\mathrm{PE}_i|}{T}. \tag{14}$$

The PE at time i (PE_i) is calculated as the difference between the prediction generated at the beginning of the trial and the actual outcome of the robot attempt to accomplish the goal (1 if the robot has achieved the goal, 0 otherwise).

ACKNOWLEDGMENT

The authors would like to thank T. Ferrauto and G. Massera for their precious help with the FARSA simulator, and O. Gigliotta for his help with statistical analysis. V. G. Santucci would like to thank T. Ferrauto, V. Sperati, F. Mannella, and D. Caligiore for their help, support, and patience.

REFERENCES

[1] M. Asada, K. F. MacDorman, H. Ishiguro, and Y. Kuniyoshi, "Cognitive developmental robotics as a new paradigm for the design of humanoid robots," *Robot. Auton. Syst.*, vol. 37, nos. 2–3, pp. 185–193, 2001.

[2] B. Bakker and J. Schmidhuber, "Hierarchical reinforcement learning based on subgoal discovery and subpolicy specialization," in *Proc. 8th Conf. Intell. Auton. Syst.*, Amsterdam, The Netherlands, 2004, pp. 438–445.

[3] G. Baldassarre, "A modular neural-network model of the basal ganglia's role in learning and selecting motor behaviours," *Cogn. Syst. Res.*, vol. 3, no. 1, pp. 5–13, 2002.

[4] G. Baldassarre, "Forward and bidirectional planning based on reinforcement learning and neural networks in a simulated robot," in *Anticipatory Behavior in Adaptive Learning Systems*. Heidelberg, Germany: Springer, 2003, pp. 179–200.

[5] G. Baldassarre, "What are intrinsic motivations? A biological perspective," in *Proc. IEEE Int. Conf. Develop. Learn. Epigenetic Robot. (ICDL-EpiRob)*, vol. 2. Frankfurt, Germany, 2011, pp. 1–8.

[6] G. Baldassarre *et al.*, "Intrinsically motivated action-outcome learning and goal-based action recall: A system-level bio-constrained computational model," *Neural Netw.*, vol. 41, pp. 168–187, May 2013.

[7] G. Baldassarre and M. Mirolli, *Computational and Robotic Models of the Hierarchical Organization of Behavior*. Berlin, Germany: Springer, 2013.

[8] G. Baldassarre and M. Mirolli, *Intrinsically Motivated Learning in Natural and Artificial Systems*. Berlin, Germany: Springer, 2013.

[9] D. H. Ballard, "Animate vision," *Artif. Intell.*, vol. 48, no. 1, pp. 57–86, 1991.

[10] A. Baranes and P.-Y. Oudeyer, "R-IAC: Robust intrinsically motivated exploration and active learning," *IEEE Trans. Auton. Mental Develop.*, vol. 1, no. 3, pp. 155–169, Oct. 2009.

[11] A. Baranes and P.-Y. Oudeyer, "Active learning of inverse models with intrinsically motivated goal exploration in robots," *Robot. Auton. Syst.*, vol. 61, no. 1, pp. 49–73, 2013.

[12] A. G. Barto, S. Singh, and N. Chantanez, "Intrinsically motivated learning of hierarchical collections of skills," in *Proc. 3rd Int. Conf. Develop. Learn. (ICDL)*, San Diego, CA, USA, 2004, pp. 112–119.

[13] A. G. Barto and S. Mahadevan, "Recent advances in hierarchical reinforcement learning," *Discrete Event Dyn. Syst.*, vol. 13, no. 4, pp. 341–379, 2003.

[14] A. G. Barto, M. Mirolli, and G. Baldassarre, "Novelty or surprise?" *Front. Psychol.*, vol. 4, pp. e1–15, Dec. 2013.

[15] A. G. Barto, R. S. Sutton, and C. W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems," *IEEE Trans. Syst., Man, Cybern.*, vol. 13, no. 5, pp. 834–846, Sep./Oct. 1983.

[16] U. R. Beierholm and P. Dayan, "Pavlovian-instrumental interaction in 'observing behavior,'" *PLoS Comput. Biol.*, vol. 6, no. 9, 2010, Art. no. e1000903.

[17] D. E. Berlyne, "Novelty and curiosity as determinants of exploratory behaviour," *Brit. J. Psychol. Gen. Sect.*, vol. 41, nos. 1–2, pp. 68–80, 1950.

[18] D. E. Berlyne, "Curiosity and exploration," *Science*, vol. 153, no. 3731, pp. 25–33, 1966.

[19] J. Boedecker, T. Lampe, and M. Riedmiller, "Modeling effects of intrinsic and extrinsic rewards on the competition between striatal learning systems," *Front. Psychol.*, vol. 4, no. 739, Oct. 2013.

[20] E. S. Bromberg-Martin and O. Hikosaka, "Midbrain dopamine neurons signal preference for advance information about upcoming rewards," *Neuron*, vol. 63, no. 1, pp. 119–126, 2009.

[21] R. A. Butler, "Discrimination learning by rhesus monkeys to visual-exploration motivation," *J. Comp. Physiol. Psychol.*, vol. 46, no. 2, pp. 95–98, 1953.

[22] D. Caligiore, M. Mirolli, D. Parisi, and G. Baldassarre, "A bioinspired hierarchical reinforcement learning architecture for modeling learning of multiple skills with continuous states and actions," in *Proc. 10th Int. Conf. Epigenetic Robot.*, vol. 149. Lund, Sweden, 2010, pp. 27–34.

[23] A. Cangelosi, M. Schlesinger, and L. B. Smith, *Developmental Robotics: From Babies to Robots*. Cambridge, MA, USA: MIT Press, 2015.

[24] E. L. Deci and R. M. Ryan, *Intrinsic Motivation and Self-Determination in Human Behavior*. New York, NY, USA: Springer, 1985.

[25] K. Doya, "Reinforcement learning in continuous time and space," *Neural Comput.*, vol. 12, no. 1, pp. 219–245, 2000.

[26] E. Duzel, N. Bunzeck, M. Guitart-Masip, and S. Duzel, "Novelty-related motivation of anticipation and exploration by dopamine (NOMAD): Implications for healthy aging," *Neurosci. Biobehav. Rev.*, vol. 34, no. 5, pp. 660–669, 2010.

[27] V. G. Fiore *et al.*, "Keep focussing: Striatal dopamine multiple functions resolved in a single mechanism tested in a simulated humanoid robot," *Front. Psychol.*, vol. 5, no. 124, pp. e1–17, 2014.

[28] W. Gerstner and W. M. Kistler, "Mathematical formulations of hebbian learning," *Biol. Cybern.*, vol. 87, nos. 5–6, pp. 404–415, 2002.

[29] H. F. Harlow, "Learning and satiation of response in intrinsically motivated complex puzzle performance by monkeys," *J. Comp. Physiol. Psychol.*, vol. 43, no. 4, pp. 289–294, 1950.

[30] T. Hester and P. Stone, "Intrinsically motivated model learning for developing curious robots," *Artif. Intell.*, May 2015, doi: 10.1016/j.artint.2015.05.002.

[31] X. Huang and J. Weng, "Novelty and reinforcement learning in the value system of developmental robots," in *Proc. 2nd Int. Workshop Epigenetic Robot. Model. Cogn. Develop. Robot. Syst.*, vol. 94. Lund, Sweden, 2002, pp. 47–55.

[32] M. Hulse, S. McBride, J. Law, and M. Lee, "Integration of active vision and reaching from a developmental robotics perspective," *IEEE Trans. Auton. Mental Develop.*, vol. 2, no. 4, pp. 355–367, Dec. 2010.

[33] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Learning attractor landscapes for learning motor primitives," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2003, pp. 1547–1554.

[34] S. Kakade and P. Dayan, "Dopamine: Generalization and bonuses," *Neural Netw.*, vol. 15, nos. 4–6, pp. 549–559, 2002.

[35] M. Khamassi, S. Lallée, P. Enel, E. Procyk, and P. F. Dominey, "Robot cognitive control with a neurophysiologically inspired reinforcement learning model," *Front. Neurorobot.*, vol. 5, no. 1, 2011, pp. 1–13.

[36] M. Khamassi, L.-E. Martinet, and A. Guillot, "Combining self-organizing maps with mixtures of experts: Application to an actor-critic model of reinforcement learning in the basal ganglia," in *From Animals to Animats 9*, S. Nolfi *et al.*, Eds. Heidelberg, Germany: Springer, 2006, pp. 394–405.

[37] G. B. Kish, "Learning when the onset of illumination is used as reinforcing stimulus," *J. Comp. Physiol. Psychol.*, vol. 48, no. 4, pp. 261–264, 1955.

[38] J. Kober and J. Peters, "Learning motor primitives for robotics," in *Proc. IEEE Int. Conf. Robot. Autom.*, Kobe, Japan, 2009, pp. 2112–2118.

[39] V. R. Kompella, M. Luciw, and J. Schmidhuber, "Incremental slow feature analysis: Adaptive low-complexity slow feature updating from high-dimensional input streams," *Neural Comput.*, vol. 24, no. 11, pp. 2994–3024, 2012.

[40] V. R. Kompella, M. Stollenga, M. Luciw, and J. Schmidhuber, "Continual curiosity-driven skill acquisition from high-dimensional video inputs for humanoid robots," *Artif. Intell.*, Feb. 2015, doi: 10.1016/j.artint.2015.02.001.

[41] G. Konidaris and A. Barto, "Skill discovery in continuous reinforcement learning domains using skill chaining," in *Adv. Neural Inf. Process. Syst. (NIPS)*, Vancouver, BC, Canada, 2009, pp. 1015–1023.

[42] G. Konidaris and A. Barto, "Efficient skill learning using abstraction selection," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, vol. 9. Pasadena, CA, USA, 2009, pp. 1107–1112.

[43] R. Lee, R. Walker, L. Meeden, and J. Marshall, "Category-based intrinsic motivation," in *Proc. 9th Int. Conf. Epigenetic Robot. (EpiRob)*, vol. 146. Venice, Italy, 2009, pp. 81–88.

[44] J. Leitner *et al.*, "Learning visual object detection and localisation using icVision," *Biol. Inspired Cogn. Architect.*, vol. 5, pp. 29–41, Jul. 2013.

[45] L. Lonini *et al.*, "Robust active binocular vision through intrinsically motivated learning," *Front. Neurorobot.*, vol. 7, no. 20, pp. e1–9, Nov. 2013.

[46] M. Lungarella, G. Metta, R. Pfeifer, and G. Sandini, "Developmental robotics: A survey," *Connect. Sci.*, vol. 15, no. 4, pp. 151–190, 2003.

[47] G. Massera, T. Ferrauto, O. Gigliotta, and S. Nolfi, "Designing adaptive humanoid robots through the FARSA open-source framework," *Adap. Behav.*, vol. 22, no. 4, pp. 255–265, 2014.

[48] A. McGovern and A. G. Barto, "Automatic discovery of subgoals in reinforcement learning using diverse density," in *Proc. 18th Int. Conf. Mach. Learn.*, Williamstown, MA, USA, 2001, pp. 361–368.

[49] K. E. Merrick, "Intrinsic motivation and introspection in reinforcement learning," *IEEE Trans. Auton. Mental Develop.*, vol. 4, no. 4, pp. 315–329, Dec. 2012.

[50] G. Metta *et al.*, "The iCub humanoid robot: An open-systems platform for research in cognitive development," *Neural Netw.*, vol. 23, nos. 8–9, pp. 1125–1134, 2010.

[51] J. H. Metzen and F. Kirchner, "Incremental learning of skill collections based on intrinsic motivation," *Front. Neurorobot.*, vol. 7, no. 11, Jul. 2013.

[52] M. Mirolli and G. Baldassarre, "Functions and mechanisms of intrinsic motivations," in *Intrinsically Motivated Learning in Natural and Artificial Systems*. Heidelberg, Germany: Springer, 2013, pp. 49–72.

[53] M. Mirolli, V. G. Santucci, and G. Baldassarre, "Phasic dopamine as a prediction error of intrinsic and extrinsic reinforcements driving both action acquisition and reward maximization: A simulated robotic study," *Neural Netw.*, vol. 39, pp. 40–51, Mar. 2013.

[54] K. C. Montgomery, "The role of the exploratory drive in learning," *J. Comp. Physiol. Psychol.*, vol. 47, no. 1, pp. 60–64, 1954.

[55] C. Moulin-Frier, S. M. Nguyen, and P.-Y. Oudeyer, "Self-organization of early vocal development in infants and machines: The role of intrinsic motivation," *Front. Psychol.*, vol. 4, no. 1006, pp. e1–20, Jan. 2014.

[56] J. Mugan and B. Kuipers, "Autonomously learning an action hierarchy using a learned qualitative state representation," in *Proc. 21st Int. Joint Conf. Artif. Intell.*, Pasadena, CA, USA, 2009, pp. 1175–1180.

[57] H. Ngo, M. Luciw, A. Förster, and J. Schmidhuber, "Confidence-based progress-driven self-generated goals for skill acquisition in developmental robots," *Front. Psychol.*, vol. 4, no. 833, pp. e1–18, Nov. 2013.

[58] M. Ogino, A. Nishikawa, and M. Asada, "A motivation model for interaction between parent and child based on the need for relatedness," *Front. Psychol.*, vol. 4, no. 618, pp. e1–11, Sep. 2013.

[59] D. Ognibene and G. Baldassare, "Ecological active vision: Four bioinspired principles to integrate bottom-up and adaptive top-down attention tested with a simple camera-arm robot," *IEEE Trans. Auton. Mental Develop.*, vol. 7, no. 1, pp. 3–25, Mar. 2015.

[60] P.-Y. Oudeyer, F. Kaplan, and V. V. Hafner, "Intrinsic motivation systems for autonomous mental development," *IEEE Trans. Evol. Comput.*, vol. 11, no. 2, pp. 265–286, Apr. 2007.

[61] P.-Y. Oudeyer and F. Kaplan, "What is intrinsic motivation? A typology of computational approaches," *Front. Neurorobot.*, vol. 1, no. 6, 2007, pp. 1–14.

[62] E. P. di Sorrentino, "Exploration and learning in capuchin monkeys (sapajus spp.): The role of action-outcome contingencies," *Anim. Cogn.*, vol. 17, no. 5, pp. 1081–1088, 2014.

[63] A. Pouget and L. H. Snyder, "Computational approaches to sensorimotor transformations," *Nat. Neurosci.*, vol. 3, pp. 1192–1198, Nov. 2000.

[64] C. Ranganath and G. Rainer, "Neural mechanisms for detecting and remembering novel events," *Nat. Rev. Neurosci.*, vol. 4, no. 3, pp. 193–202, 2003.

[65] P. Redgrave and K. Gurney, "The short-latency dopamine signal: A role in discovering novel actions?" *Nat. Rev. Neurosci.*, vol. 7, no. 12, pp. 967–975, 2006.

[66] M. Rolf and M. Asada, "Autonomous development of goals: From generic rewards to goal and self detection," in *Proc. 4th Joint IEEE Int. Conf. Develop. Learn. Epigenetic Robot. (ICDL-Epirob)*, Genoa, Italy, 2014, pp. 187–194.

[67] M. Rolf, J. J. Steil, and M. Gienger, "Goal babbling permits direct learning of inverse kinematics," *IEEE Trans. Auton. Mental Develop.*, vol. 2, no. 3, pp. 216–229, Sep. 2010.

[68] M. Rolf, J. J. Steil, and M. Gienger, "Online goal babbling for rapid bootstrapping of inverse models in high dimensions," in *Proc. 1st Joint IEEE Int. Conf. Develop. Learn. Epigenetic Robot. (ICDL-Epirob)*, vol. 2. Frankfurt, Germany, 2011, pp. 1–8.

[69] E. T. Rolls, A. Treves, and E. T. Rolls, *Neural Networks and Brain Function*. New York, NY, USA: Oxford Univ. press, 1998.

[70] R. M. Ryan and E. L. Deci, "Intrinsic and extrinsic motivations: Classic definitions and new directions," *Contemp. Educ. Psychol.*, vol. 25, no. 1, pp. 54–67, 2000.
[71] V. G. Santucci, G. Baldassarre, and M. Mirolli, "Intrinsic motivation mechanisms for competence acquisition," in *Proc. 2nd Joint IEEE Int. Conf. Develop. Learn. Epigenetic Robot. (ICDL-Epirob)*, San Diego, CA, USA, 2012, pp. 1–6.
[72] V. G. Santucci, G. Baldassarre, and M. Mirolli, "Autonomous selection of the "what" and the "how" of learning: An intrinsically motivated system tested with a two armed robot," in *Proc. 4th Joint IEEE Int. Conf. Develop. Learn. Epigenetic Robot. (ICDL-Epirob)*, Genoa, Italy, 2014, pp. 434–439.
[73] V. G. Santucci, G. Baldassarre, and M. Mirolli, "Intrinsic motivation signals for driving the acquisition of multiple tasks: A simulated robotic study," in *Proc. 12th Int. Conf. Cogn. Model. (ICCM)*, Ottawa, ON, Canada, 2013, pp. 59–64.
[74] V. G. Santucci, G. Baldassarre, and M. Mirolli, "Which is the best intrinsic motivation signal for learning multiple skills?" *Front. Neurorobot.*, vol. 7, no. 22, pp. e1–14, Nov. 2013.
[75] V. G. Santucci, G. Baldassarre, and M. Mirolli, "Biological cumulative learning through intrinsic motivations: A simulated robotic study on the development of visually-guided reaching," in *Proc. 10th Int. Conf. Epigenetic Robot. (EpiRob)*, vol. 149. Lund, Sweden, 2010, pp. 121–127.
[76] M. Schembri, M. Mirolli, and G. Baldassarre, "Evolving childhood's length and learning parameters in an intrinsically motivated reinforcement learning robot," in *Proc. 7th Int. Conf. Epigenetic Robot.*, Lund, Sweden, 2007, pp. 141–148.
[77] M. Schembri, M. Mirolli, and G. Baldassarre, "Evolving internal reinforcers for an intrinsically motivated reinforcement-learning robot," in *Proc. 6th Int. Conf. Develop. Learn.*, London, U.K., 2007, pp. 282–287.
[78] J. Schmidhuber, "Curious model-building control system," in *Proc. Int. Joint Conf. Neural Netw.*, vol. 2. Singapore, 1991, pp. 1458–1463.
[79] J. Schmidhuber, "A possibility for implementing curiosity and boredom in model-building neural controllers," in *Proc. Int. Conf. Simulat. Adap. Behav. Anim. Animats*, Cambridge, MA, USA, 1991, pp. 222–227.
[80] J. Schmidhuber, "Formal theory of creativity, fun, and intrinsic motivation (1990-2010)," *IEEE Trans. Auton. Mental Develop.*, vol. 2, no. 3, pp. 230–247, Sep. 2010.
[81] W. Schultz, "Predictive reward signal of dopamine neurons," *J. Neurophysiol.*, vol. 80, no. 1, pp. 1–27, 1998.
[82] A. Stout and A. G. Barto, "Competence progress intrinsic motivation," in *Proc. 9th IEEE Int. Conf. Develop. Learn. (ICDL)*, Ann Arbor, MI, USA, 2010, pp. 257–262.
[83] F. Stulp, E. A. Theodorou, and S. Schaal, "Reinforcement learning with sequences of motion primitives for robust manipulation," *IEEE Trans. Robot.*, vol. 28, no. 6, pp. 1360–1370, Dec. 2012.
[84] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.
[85] R. S. Sutton, D. Precup, and S. Singh, "Between MDPS and semi-MDPS: A framework for temporal abstraction in reinforcement learning," *Artif. Intell.*, vol. 112, pp. 181–211, Aug. 1999.
[86] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2553–2561.
[87] F. Taffoni *et al.*, "Development of goal-directed action selection guided by intrinsic motivations: An experiment with children," *Exp. Brain Res.*, vol. 232, no. 7, pp. 2167–2177, 2014.
[88] P. Tommasino, D. Caligiore, M. Mirolli, and G. Baldassarre, "Reinforcement learning algorithms that assimilate and accommodate skills with multiple tasks," in *Proc. IEEE Int. Conf. Develop. Learn. Epigenetic Robot. (ICDL)*, San Diego, CA, USA, 2012, pp. 1–8.
[89] P. Tommasino, D. Caligiore, M. Mirolli, and G. Baldassarre, "Transfer expert reinforcement learning (TERL): A reinforcement learning architecture that transfers knowledge between skills," *IEEE Trans. Auton. Mental Develop.*, to be published.
[90] C. M. Vigorito and A. G. Barto, "Intrinsically motivated hierarchical skill learning in structured environments," *IEEE Trans. Auton. Mental Develop.*, vol. 2, no. 2, pp. 132–143, Jun. 2010.
[91] C. V. Hofsten, "An action perspective on motor development," *Trends Cogn. Sci.*, vol. 8, no. 6, pp. 266–272, 2004.
[92] J. Weng *et al.*, "Autonomous mental development by robots and animals," *Science*, vol. 291, no. 5504, pp. 599–600, 2001.
[93] R. W. White, "Motivation reconsidered: The concept of competence," *Psychol. Rev.*, vol. 66, pp. 297–333, Sep. 1959.
[94] L. Wiskott and T. J. Sejnowski, "Slow feature analysis: Unsupervised learning of invariances," *Neural Comput.*, vol. 14, no. 4, pp. 715–770, 2002.
[95] B. C. Wittmann, N. D. Daw, B. Seymour, and R. J. Dolan, "Striatal activity underlies novelty-based choice in humans," *Neuron*, vol. 58, no. 6, pp. 967–973, 2008.

Vieri Giuliano Santucci received the B.Sc. degree in philosophy from the University of Pisa, Pisa, Italy, in 2006, and the M.S. degree in theories and techniques of knowledge from the Faculty of Philosophy, University of Rome "La Sapienza," Rome, Italy, in 2009, and the Ph.D. degree in computer science from the University of Plymouth, Plymouth, U.K., in 2016, with a focus on the development of robotic architectures that allow artificial agents to autonomously improve their competences on the basis of the biologically-inspired concept of intrinsic motivations.

He is a Research Assistant with the Istituto di Scienze e Tecnologie della Cognizione, Consiglio Nazionale delle Ricerce, Rome. He published in peer-reviewed journals and attended several international conferences, and actively contributed to the European Integrated Project "IM-CLeVeR—Intrinsically-Motivated Cumulative-Learning Versatile Robots." His current research interests include learning processes to motivations as well as to the concept of representations, in both biological and artificial agents.

Gianluca Baldassarre received the B.A. and M.A. degrees in economics and the M.Sc. degree in cognitive psychology and neural networks from the University of Rome "La Sapienza," Rome, Italy, in 1998 and 1999, respectively, and the Ph.D. degree in computer science with the University of Essex, Colchester, U.K., in 2003, with a focus on planning with neural networks.

He was a Post-Doctoral Fellow with the Italian Institute of Cognitive Sciences and Technologies, National Research Council, Rome, researching on swarm robotics, where he has been a Researcher, since 2006, and coordinates the Research Group that he founded called the Laboratory of Computational Embodied Neuroscience. From 2006 to 2009, he was a Team Leader of the EU Project "ICEA—Integrating Cognition Emotion and Autonomy" and the Coordinator of the European Integrated Project "IM-CLeVeR—Intrinsically-Motivated Cumulative-Learning Versatile Robots," from 2009 to 2013. He has over 100 international peer-review publications. His current research interests include cumulative learning of multiple sensorimotor skills driven by extrinsic and intrinsic motivations. He studies these topics with two interdisciplinary approaches: with computational models constrained by data on brain and behavior, aiming to understand the latter ones and with machine-learning/robotic approaches, aiming to produce technologically useful robots.

Marco Mirolli received the M.S. degree in philosophy from the University of Siena, Siena, Italy, in 2001, and the Ph.D. degree in cognitive sciences, Siena, in 2006.

He is a Researcher with the Istituto di Scienze e Tecnologie della Cognizione, Consiglio Nazionale delle Ricerce, Rome, Italy. Throughout his career, he has been (mostly) studying brain and behavior through computer simulations and robotic models. In particular, he has been researching on the evolution of communication and language, the role of language as a cognitive tool, the concept of representations in cognitive science, intrinsic motivations, and the biological bases of conditioning, motivations, and emotions. He has co-edited four books, including *Intrinsically Motivated Learning in Natural and Artificial Systems* (Springer) and published over 60 peer-reviewed papers. His current research interests include understanding the relationships between the body and the mind through theoretical analysis, computational modeling, and empirical research.